JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

1/2025

vol. 99

Enhancing Phishing Detection in Cloud Environments Using RNN-LSTM in a Deep Learning Framework

Oussama Senouci and Nadjib Benaouda 1

Cat Swarm Optimization with Lévy Flight for Link Load Balancing in SDN

Kwaku Kwarteng, Kwame O. Gyasi, Justice O. Agyemang, Kwame Agyekum, et al. 10

FPGA-based Low Latency Square Root CORDIC Algorithm

Mariusz Węgrzyn, Stepan Voytusik, and Nataliia Gavkalova 21

Task Offloading and Scheduling Based on Mobile Edge Computing and Software-defined Networking

Fatimah Azeez Rawdhan 30

A Hole-free Shifted Coprime Array for DOA Estimation

Fatimah Abdulnabi Salman and Bayan Mahdi Sabbar 38

Context-Awareness for Device-to-Device Resource Allocation

Marcin Rodziewicz 47

Semantic Segmentation of Plant Structures with Deep Learning and Channel-wise Attention Mechanism

Mukund Kumar Surehli, Naveen Aggarwal, Garima Joshi, and Harsh Nayyar 56

Ultra-wideband Antenna System Design for Future mmWave Applications

Muhannad Y. Muhsin, Zainab S. Muqdad, Asaad H. Sahar, Zainab F. Mohammad, et al. 67

(*Contents continued on back cover*)

eISSN 1899-8852

Enhancing Phishing Detection in Cloud Environments Using RNN-LSTM in a Deep Learning Framework

Oussama Senouci and Nadjib Benaouda

University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria

https://doi.org/10.26636/jtit.2025.1.1916

Abstract — Phishing attacks targeting cloud computing services are more sophisticated and require advanced detection mechanisms to address evolving threats. This study introduces a deep learning approach leveraging recurrent neural networks (RNNs) with long short-term memory (LSTM) to enhance phishing detection. The architecture is designed to capture sequential and temporal patterns in cloud interactions, enabling precise identification of phishing attempts. The model was trained and validated using a dataset of 10,000 samples, adapted from the PhishTank repository. This dataset includes a diverse range of attack vectors and legitimate activities, ensuring comprehensive coverage and adaptability to real-world scenarios. The key contribution of this work includes the development of a high-performance RNN-LSTM-based detection mechanism optimized for cloud-specific phishing patterns that achieve 98.88% accuracy. Additionally, the model incorporates a robust evaluation framework to assess its applicability in dynamic cloud environments. The experimental results demonstrate the effectiveness of the proposed approach, surpassing existing methods in accuracy and adaptability.

Keywords — cloud services, cybersecurity, deep learning, phishing detection, RNN-LSTM

1. Introduction

Phishing attacks have emerged as one of the most persistent cybersecurity threats, exploiting a combination of social engineering and technical manipulation to trick users into revealing sensitive information such as passwords, financial data, and personal identification details [1]. The widespread adoption of cloud computing services for data storage, processing, and communication has made these platforms a target for attackers. Cybercriminals are increasingly designing phishing campaigns that resemble legitimate communications from cloud service providers, leveraging authentic user interfaces to gain user trust and compromise security [2]. This growing threat highlights the need for advanced detection mechanisms tailored to the dynamic nature of the cloud environment.

The sophistication of phishing attacks in cloud ecosystems poses significant challenges to traditional detection methods. Approaches such as URL blacklisting and heuristic-based systems struggle to keep up with evolving attack techniques [3]. Additionally, the rapid evolution of cloud services, coupled with their diverse applications, complicates the task of identifying malicious activities. Attackers often exploit multiple channels, including email, social networks, and messaging platforms, to deceive users. Furthermore, the rise of phishing-as-a-service (PhaaS) has simplified the process for cybercriminals, enabling large-scale, highly tailored phishing campaigns with minimal effort [4], [5].

To address these challenges, machine learning (ML) and deep learning (DL) techniques have proven to be effective tools for phishing detection. Among these, recurrent neural networks (RNNs) combined with long- and short-term memory units (LSTM) have demonstrated exceptional capability in analyzing sequential data and capturing long-term dependencies. This architecture is particularly well suited to detect behavioral anomalies in cloud-based systems, as it dynamically adapts to temporal patterns in user interactions [6].

This study focuses on leveraging the RNN-LSTM architecture to enhance phishing detection by analyzing irregular access behaviors, suspicious data transfers, and deceptive communications in cloud environments [7]. The presented research uses the publicly available PhishTank dataset, which has been adapted to simulate cloud-specific phishing scenarios. Although the dataset was not created from scratch, it has been tailored to include a diverse range of phishing attacks targeting cloud platforms. Examples include fake login pages, malicious links embedded in cloud communications, and phishing attempts to exploit cloud-based file-sharing services. By refining this dataset to focus on cloud-relevant attack vectors, the study bridges the gap between general phishing detection methods and the challenges posed in cloud environments.

The key contributions of this work are as follows:

1) design and implementation of an RNN-LSTM-based deep learning architecture, optimized to capture sequential and temporal patterns in cloud service interactions. This architecture enhances the detection of subtle anomalies indicative of phishing activities while maintaining computational efficiency.
2) adaptation of the PhishTank data set to reflect specific real-world phishing scenarios for cloud platforms, ensuring comprehensive representation of various attack strategies.
3) development of a robust detection mechanism that integrates RNN-LSTM capabilities with adjustments for

cloud-specific challenges, achieving high accuracy and adaptability across various cloud environments.

4) design of a comprehensive evaluation framework to assess system performance under real-world conditions, focusing on metrics such as detection accuracy, false positives, and adaptability to emerging threats.
5) introduction of enhanced protection strategies aimed at strengthening cloud security by mitigating the risks associated with phishing attacks and reducing potential data breaches and financial losses.

The remainder of this paper is structured as follows: Section 2 reviews related work on phishing detection with a particular focus on cloud-specific challenges and limitations. Section 3 details the proposed RNN-LSTM-based methodology and the adaptation of the dataset. Section 4 presents the experimental results and evaluates the performance of the detection system. Finally, Section 5 concludes the paper by summarizing the contributions and discussing directions for future research.

2. Literature Review

The issue of phishing in cloud environments has gained significant attention from researchers, leading to various approaches aimed at improving detection mechanisms. Existing studies provide a foundation for the methodology adopted in this work by exploring trends, vulnerabilities, and specific detection strategies for phishing attacks in cloud infrastructures. Traditional approaches, such as blacklisting URLs and rule-based systems, have proven effective to some extent, but struggle to keep pace with evolving phishing techniques. The application of deep learning and advanced pattern recognition methods has emerged as a promising alternative, offering improved accuracy and adaptability.

In previous studies, a variety of datasets containing legitimate and phishing-related data have been utilized in previous studies. For example, the authors of [8] proposed a phishing detection system that uses email data and employs algorithms such as support vector machines (SVM), naive Bayes (NB) and LSTM. Their method involves extracting features from emails and creating labeled datasets for classification. Although effective in identifying phishing patterns, this approach relies heavily on predefined features, making it less adaptable to emerging attack strategies.

In study [9], the authors developed a phishing email detection model based on deep learning techniques, such as graph convolutional networks (GCNs) combined with natural language processing. By focusing on the textual content of emails, the system demonstrated improved detection accuracy. However, the reliance on annotated training data presents challenges in scalability, as the manual labeling of large datasets can be time and resource consuming.

A different approach was taken in [10], where a data-driven model was proposed to detect phishing web pages using a multilayer perceptron (MLP). This study utilized a Kaggle dataset comprising 10 features and 10,000 websites, achieving training and testing accuracies of 95% and 93%, respectively. Despite these promising results, the model's dependence on limited features restricts its ability to generalize to phishing web pages employing novel evasion techniques.

The work [11] introduced PhishNot, a system for detecting phishing URLs using machine learning. By reducing the input features to 14, the authors optimized the system for fast processing and demonstrated a high detection accuracy of 97.5% using a random forest algorithm. Although this streamlined approach is computationally efficient, it may fail to capture complex patterns in sophisticated phishing URLs, potentially limiting its effectiveness in more advanced scenarios.

In article [12], a hybrid CNN-LSTM model was proposed to detect phishing attacks through image-based encoding. The system incorporated advanced techniques such as SMOTE and auto-encoder-based GANs to balance datasets and extract meaningful features. Grayscale images were used for the analysis and the approach achieved superior performance on multiple benchmarks. However, the computational complexity of this method may pose challenges for real-time detection or deployment in resource-constrained environments.

Collectively, these studies highlight the potential and limitations of existing phishing detection techniques. Based on their findings, this research aims to address key gaps by developing a more adaptable and scalable system to detect phishing activities in cloud-hosted environments.

2.1. Comparison and Limitations of Existing Methods

Table 1 presents a comparison of the phishing detection approaches reviewed in this study. It describes the type of model, training data used, accuracy achieved, complexity of the model, robustness to evolving phishing tactics, processing time, and key limitations associated with each method. This comparative analysis highlights the strengths and weaknesses of the existing approaches, offering valuable insights into their real-world applicability. By evaluating these characteristics, one can identify the most suitable phishing detection strategy for their specific needs.

Despite notable progress in phishing detection, the following limitations are evident in current methods:

1) Dependence on predefined features. The approach [8] is based on fixed set of characteristics, which may not effectively capture the nuances of emerging phishing strategies. Frequent updates are necessary to maintain its relevance in detecting advanced attacks.
2) Requirement for annotated training data. The method presented in [9] uses supervised learning, which requires large amounts of annotated data. This dependency increases the cost and effort required, limiting scalability and adaptability to novel phishing techniques.
3) Narrow scope and limited features. The framework [10] employs an MLP trained on a dataset with only ten attributes. This restricts the ability to handle phishing web pages using innovative tactics or evasion methods.
4) Simplified feature sets at the expense of complexity. The PhishNot system [11] reduces the input features to only 14

Tab. 1. Comparison of phishing detection approaches.

Approach	Model type	Training data	Accuracy	Complexity	Robustness	Processing time
[8]	SVM, NB, LSTM	Various datasets	95% (estimated)	Medium	Low	Fast
[9]	GCN	Annotated email data	93% (test)	High	Medium	Moderate
[10]	MLP	10,000 webpages	95% (training)	High	Low	Moderate
[11]	Random forest	Representative data	97.5%	Medium	Low	Fast
[12]	CNN-LSTM	3 benchmark datasets	Superior to others	High	Low	Slow

to achieve faster processing. However, this simplification may exclude intricate phishing URL patterns, reducing its effectiveness against sophisticated attacks.

5) High computational overhead. The CNN-LSTM approach shown in [12] involves advanced techniques such as SMOTE, GANs, and swarm intelligence, significantly increasing computational requirements. This complexity makes it unsuitable for real-time detection or resource-constrained environments.

3. Proposed Approach

In this section, we introduce a deep learning-based approach for detecting phishing in cloud environments leveraging an RNN-LSTM model. The proposed system is specifically designed to address the dynamic and evolving nature of phishing attacks in cloud settings by capturing sequential and temporal patterns in user interactions. The methodology consists of four phases: data acquisition and preprocessing, feature representation, model training, and performance evaluation.

The first phase involves gathering a comprehensive dataset comprising legitimate and phishing interactions in cloud services. The PhishTank[1]. dataset is curated to ensure its relevance to real-world scenarios. To prepare the dataset for analysis, it undergoes pre-processing steps, including cleaning to remove inconsistencies, normalization to standardize data attributes, and the extraction of relevant features such as URL length, IP address presence, and HTTPS usage. This ensures that the data set is high quality and suitable for deep learning.

The second phase focuses on transforming the data into a format compatible with the RNN-LSTM architecture. Textual input, such as URLs, are converted to numerical representations using character-level embeddings. These embeddings preserve contextual relationships within the data, enabling the model to analyze sequential patterns effectively and detect phishing behaviors.

The third phase is dedicated to training the RNN-LSTM model. The architecture is designed to capture temporal dependencies in the data, using LSTM layers to retain essential information over time. Regularization techniques, such as dropout, are applied during training to prevent overfitting and improve the generalization of the model. Hyperparameters, including the learning rate, batch size, and the number of LSTM units, are optimized through systematic grid search and cross-validation.

Finally, the model performance is evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics assess the system's effectiveness in distinguishing phishing attempts from legitimate interactions. The analysis also includes an evaluation of the importance of individual characteristics in the detection process, providing information on the decision-making and the patterns it identifies as indicative of phishing behavior.

3.1. Data Exploration and Analysis

PhishTank, a community-curated repository of validated phishing websites, serves as the main data source for this study. This database aggregates reports of suspicious websites from users and validates their authenticity through expert review. Each entry includes critical information such as the URL of the phishing site's URL, submission time, verification status, and operational state. PhishTank provides these data in both API and CSV formats, making them highly accessible for machine learning research [13]. For this investigation, we used 10,000 samples, evenly categorized as authentic or phishing based on 18 distinct attributes that focus on URL structure and activity patterns. These attributes enable a detailed analysis of phishing activity, including detection patterns, response times, and the lifecycle of malicious URLs.

A preliminary exploration of the dataset revealed patterns in phishing behavior and response efficiency. Key observations include:

- **Phishing lifecycle**. The dataset records both submission and verification timestamps, allowing for analysis of the time taken to confirm a phishing report. This provides insight into response efficiency.
- **URL characteristics**. Phishing URLs often feature irregular structures, including excessive length, presence of IP addresses, or frequent use of special characters, which differentiate them from legitimate URLs.

[1] The PhishTank dataset is accessible at https://phishtank.org

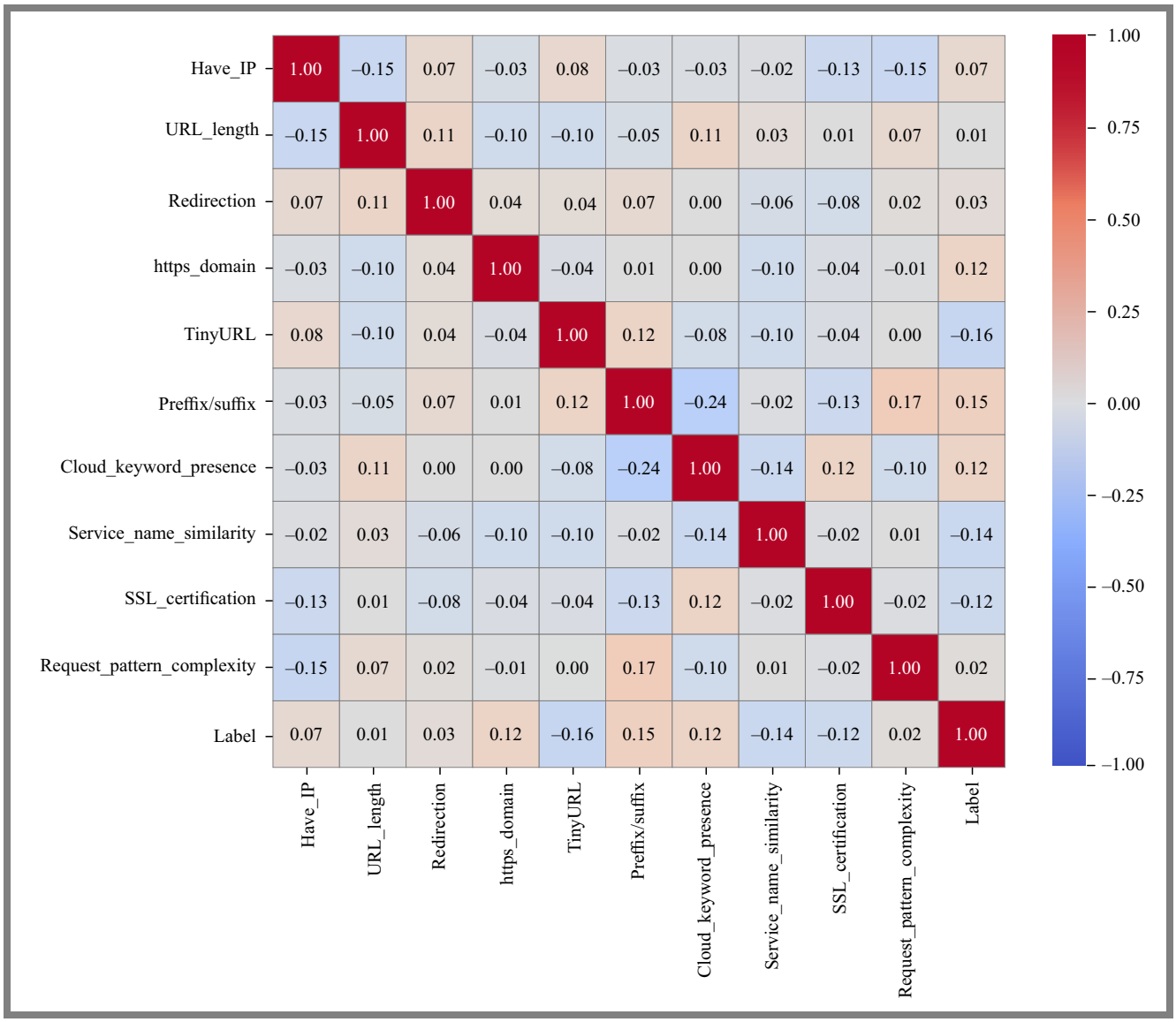

Fig. 1. Correlation matrix of features.

- **Verification patterns**. Most verified phishing sites are confirmed within a short time frame, highlighting the effectiveness of the PhishTanks validation process.

To ensure the quality of the dataset for training and evaluation, the following steps were undertaken:

- removal of incomplete or duplicate entries to maintain data integrity,
- normalization of URL features to ensure consistency across samples,
- label encoding of target variables, with phishing URLs labeled as "1" and legitimate URLs as "0".

These steps ensure that the dataset is both clean and ready for robust feature extraction and model training.

3.2. Feature Extraction

To ensure robust detection, we extracted significant features relevant to identifying phishing attempts in cloud environments. The following key attributes, aligned with our study and correlation analysis, are particularly essential:

– Having_IP – indicates the presence of an IP address in the URL, a tactic commonly used in phishing attacks to bypass domain-based detection.
– URL_Length – represents the total length of the URL. Phishing URLs often have unusually long paths to obfuscate their content.
– Redirection – counts the number of redirects in a URL. Phishing websites frequently use multiple redirections to conceal their malicious intent.
– Https_Domain – verifies the presence of HTTPS in the domain. Phishing attackers often exploit this to create a false sense of security.
– TinyURL – identifies the use of URL shortening services, which are frequently leveraged in phishing campaigns to obscure malicious destinations.

– Prefix/Suffix – checks for the presence of prefix or suffix patterns in domain names, as these can mimic legitimate domains.
– Cloud_Keyword_Presence – detects cloud-related keywords (e.g., cloud, aws, azure) in the URL, which phishers may use to impersonate trusted cloud providers.
– Service_Name_Similarity – analyzes the similarity between the URL content and popular cloud service names to deceive users into assuming legitimacy.
– SSL_Certification – determines the validity of SSL certificates as phishing websites may use SSL to appear credible.
– Request_Pattern_Complexity – evaluates the complexity of the URL's request pattern, as phishing sites often include irregular or unusually structured paths.
– Label – the target variable, classifying URLs as phishing or legitimate.

These features are essential for identifying phishing websites by analyzing both structural and behavioral patterns. Their correlations, as shown in Fig. 1, highlight their significance in cloud phishing detection models in the form of a correlation matrix.

Each dataset feature is labeled 1 for phishing websites and 0 for legitimate ones. Due to the size (20,000 entries: 10,000 phishing URLs and 10,000 authentic), seven significant features were extracted. These qualities help identify legitimate and fraudulent websites – see Fig. 2.

3.3. Proposed Phishing Detection Model

This subsection introduces the architecture and functionality of the proposed phishing detection model. The model leverages an RNN framework augmented with LSTM layers to effectively capture sequential and temporal patterns in cloud service interactions. By analyzing these behavioral patterns over time, the model can differentiate between phishing and legitimate activities with high precision.

The proposed architecture is designed to process input sequences while preserving contextual and temporal dependencies critical for identifying phishing attempts. The model is made up of multiple layers, each serving a specific role in the detection pipeline.

In the embedding layer, input tokens are first converted into dense vectors that represent their semantic meaning. This layer is crucial for the model to understand the contextual links. The expression of embedding transformation is as follows:

$$\mathbf{E}(X) = \{e(x_1), e(x_2), \ldots, e(x_T)\} \,, \tag{1}$$

where $e(x_i)$ is the embedding vector for token x_i.

Embedding tokens into dense representations gives the model a numerical representation of text that improves interpretability for future layers, helping it uncover phishing patterns. The quality of the embeddings can be improved using pre-trained models like Word2Vec or GloVe, which can be integrated into the embedding layer in such a way:

$$e(x_i) = \mathbf{W} \cdot v_i \,, \tag{2}$$

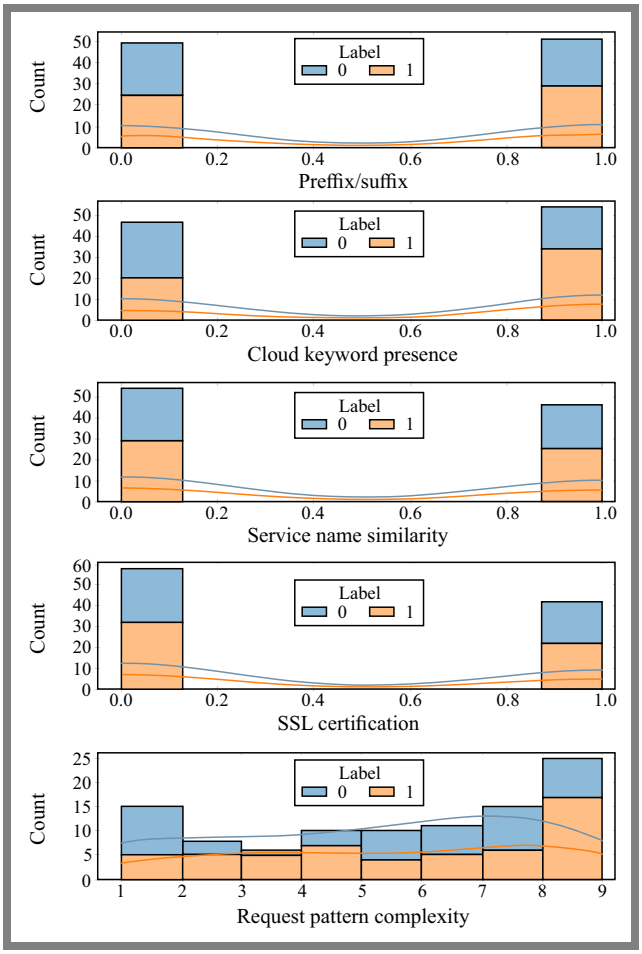

Fig. 2. Histogram of feature visualization.

where $\mathbf{W}$ is the weight matrix and v_i is the vector representation of the pre-trained model.

The LSTM layers capture embedded data sequence dependencies. LSTMs can keep crucial information and discard less important information, making them suitable for detecting phishing patterns that depend on earlier or later sequences. Each LSTM layer computes hidden states h_t and cell states C_t at each time step t, as defined by:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(forget gate)} \,, \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(input gate)} \,, \tag{4}$$

$$\tilde{C}_t = \text{tgh}(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{(cell candidate)} \,, \tag{5}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad \text{(cell state)} \,, \tag{6}$$

$$h_t = o_t \cdot \text{tgh}(C_t) \quad \text{(hidden state)} \,. \tag{7}$$

Eqs. (3) – (7), $f_t, i_t, C_t, \tilde{C}_t, h_t$, and o_t represent the forget gate, the input gate, cell candidate, the cell state the hidden state, and output gate, respectively.

The LSTM technique selectively updates and preserves cell states through these gates to keep the model's phishing detection knowledge. To enhance the model's capacity to capture

long-range dependencies, we can also implement a bidirectional LSTM, which processes the input sequence in both forward and backward directions.

$$h_t = \text{LSTM}(x_t, h_{t-1}, C_{t-1}) + \text{LSTM}(x_t^{\text{rev}}, h_{t-1}^{\text{rev}}, C_{t-1}^{\text{rev}}) \, . \quad (8)$$

A fully connected layer and output layer evaluate the final hidden state h_T after LSTM layers, converting the learned features into a probability score for phishing attempts. The prediction formula is:

$$\hat{y} = \sigma(W_O \cdot h_T + b_O) \, , \quad (9)$$

where σ denotes sigmoid activation.

The probability of the sequence being phishing is between 0 and 1 with this function. These layers decide by mapping the RNN's learned properties into a binary outcome. Next, the model is trained by minimizing a binary cross-entropy loss function that compares true labels and predicted probability. The loss function encourages accurate model predictions by reducing the error between true labels and predictions. The definition of a loss function is as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right) \, , \quad (10)$$

where, y_i represents the true label, while $\hat{y}_i$ denotes the predicted probability for each instance.

Regularization methods such as dropout can be added after LSTM layers to enhance generalization and reduce overfitting. The dropout layer allows the model to avoid over-reliance on any single feature by randomly setting a percentage p of input units to zero during training. A mathematical expression is:

$$h_t^{\text{drop}} = h_t \cdot r \, , \quad (11)$$

where r is a random variable drawn from a Bernoulli distribution with parameter p.

As evaluation metrics, precision, recall, and F1 score measures are used to assess the model's performance. The F1 score is calculated as follows:

$$\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \, . \quad (12)$$

Building a robust phishing detection system for cloud environments requires thorough training and meticulous tuning to adapt to the constantly evolving nature of phishing threats.

The performance of the model is optimized through careful selection of hyper parameters, including the learning rate, batch size, number of LSTM layers, and dropout rate. Hyperparameter tuning is performed systematically using grid search and cross-validation to identify the best configuration. Regularization techniques are applied to mitigate overfitting, ensuring that the model maintains strong predictive capabilities when exposed to unseen data.

The following parameters are assumed during model training:

– Embedding dimension = 256, enhancing the representation of tokens to capture more intricate semantic relationships while still being manageable in terms of computational resources.
– LSTM units = 256 for each LSTM layer, increasing the model's ability to effectively learn and retain complex sequential dependencies in the data.
– Dropout rate = 0.3, striking a balance between reducing overfitting and preserving enough model capacity to ensure robust learning.
– Batch size = 64, allowing for more stable gradient updates while also accommodating larger data chunks, which can lead to better convergence.
– Epochs = 50, with an emphasis on using early stopping based on validation loss to ensure that the model does not overtrain while still having sufficient training time to achieve high accuracy.

The computationally effective embedding dimension of 256 captures more intricate semantic links and improves token representation. Each 256-unit LSTM layer helps the model learn and retain complicated sequential data dependencies. A dropout rate of 0.3 balances overfitting with learning model robustness. With a batch size of 64, larger training data chunks improve gradient updates and convergence. To avoid overtraining and ensure accuracy, the model is trained for more than 50 epochs and stopped when validation loss occurs.

4. Performance Evaluation

The proposed model was built and tested using Google Colab, a powerful cloud-based machine learning model deployment and testing tool. Free GPU resources make Google Colab ideal for testing complex algorithms and datasets.

The metrics considered during the evaluation of the proposed approach include:

- **Accuracy**, comparing predicted labels to actual labels gives a clear picture of the model performance.
- **Precision, recall and F1 score**. Precision assesses a model's ability to correctly identify affirmative phishing attempts, while recall measures its ability to catch all actual incidents. The F1 score is a harmonic mean of precision and recall that accounts for false positives and negatives, which is essential to analyze model performance in imbalanced datasets.
- **Confusion matrix**. This tool provides a detailed breakdown of the model's classification performance by showing the true positives, true negatives, the false positives, and false negatives. Analyzing the confusion matrix allows for a deeper understanding of specific weaknesses in the model, such as misclassifying legitimate URLs as phishing attempts, which can be particularly detrimental in practical applications.
- **Loss function**. In the context of RNNs, monitoring the loss function during training is vital for assessing model performance. A decreasing loss value indicates that the model is learning effectively, while fluctuations can signal over- or underfitting issues.

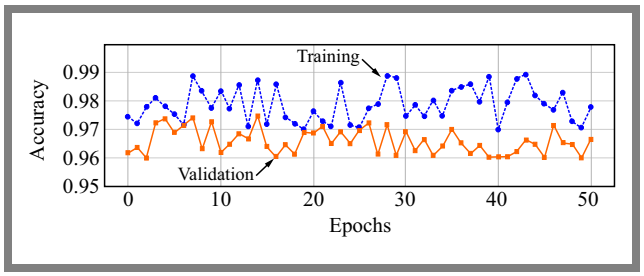

Fig. 3. Model training and validation accuracy performance.

4.1. Results Analysis

Figure 3 illustrates the performance metrics of the RNN deep learning model used to detect phishing attempts in cloud-based applications. The algorithm consistently identifies complex traits that separate phishing interactions from real ones, with training accuracy between 97% and 99%. The model validation accuracy varies from 96% to 97.5%, demonstrating its ability to generalize to unknown data, crucial for real-world applications.

Training loss of optimization targets, the difference between predicted outputs and targets. Low validation loss indicates learning efficacy, while low training loss indicates that model predictions meet goals. Data pattern capture issues may cause high validation loss, while good learning indicates low validation loss.

The model may overfit or underfit if it memorizes the training data or is too basic to detect patterns. Tracking training loss is essential. The robust model has 98.885% training accuracy and 97.75% validation accuracy, indicating that there are no overfits or underfitting. The model recognizes substantial training data patterns and generalizes well to novel samples. Real-world phishing detection requires balanced models with low false positives. Consistent RNN performance improves cloud phishing security.

Figure 4 shows the RNN deep learning model loss performance metrics to identify phishing attempts in cloud-based applications. The model minimizes the difference between expected outputs and targets with a training loss of 0.025 to 0.038. Reduced training loss shows that the model can handle complicated data patterns. The validation loss is 0.053 to 0.068, indicating a strong new data generalization. The model finds fundamental patterns without overfitting the training dataset, since validation loss is always less than training loss.

Training and validation loss might show overfitting and underfitting, when the model memorizes training data but misses data patterns. The resilience balances detail and generalization with 0.031 training loss and 0.060 validation loss. Phishing detection models with low false positives must balance performance. Loss performance measurements show RNN architecture stability, securing cloud from phishing.

Figure 5 shows the confusion matrix for the proposed model, assessing its classification performance. This matrix indicates the distinction and proposes improvements. High false negatives signal that the model must be modified to detect more positives. Overall, the balanced true positives and true negatives of the matrix show that the program could detect

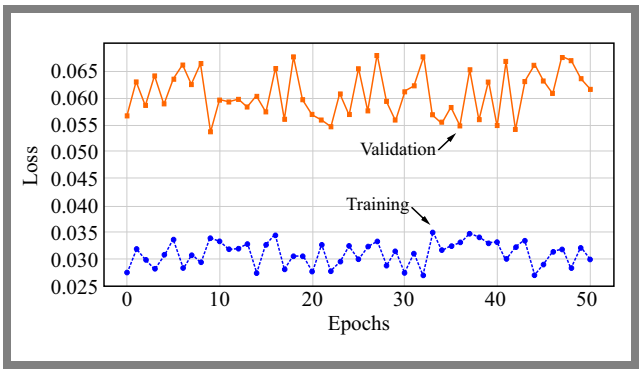

Fig. 4. Performance of the training and validation loss of proposed model.

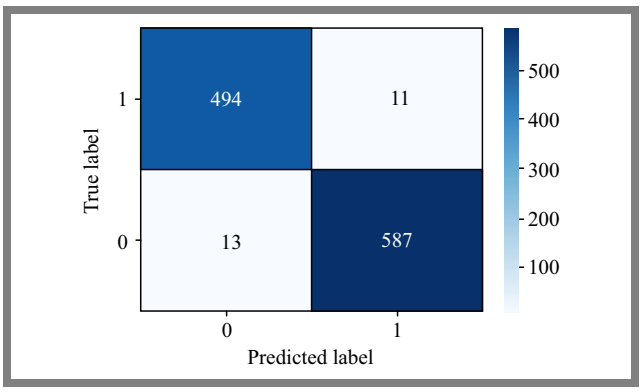

Fig. 5. Confusion matrix for the proposed model.

phishing attempts. This robust result shows that the proposed model is acceptable for real-world applications with precision enhancement potential.

Table 2 summarizes model's evaluation metrics. The model classifies the dataset samples well with 0.988 accuracy. The macro- and micro-averaged precision and recall indicate positive sample detection with low false positives. The recall scores and macro- and microaveraged precision of 0.979 and 0.982 demonstrate its ability to recover relevant samples. The methodology works because macro and micro F1 scores average precision and recall into a harmonic mean. The macro-averaged F1 score of 0.977 and micro-averaged F1 score of 0.984 indicate balanced and trustworthy performance. These results show that the proposed approach can generalize and reduce classification errors for real-world applications.

Table 3 shows the accuracy and loss metrics for model training and validation. The high training accuracy of 0.988 and the validation accuracy of 0.977 show the model ability to capture patterns in the training data and generalize to new data. A low training loss of 0.0318 shows minimal inaccuracy in predicting training samples, while a validation loss of 0.0537 indicates consistent performance on unseen data without over- or underfitting. These indicators demonstrate the stability and learning potential, promising applications in the real world.

Table 4 analyzes the accuracy of the model using PhishTank dataset, proving its effectiveness in phishing detection. The RNN-LSTM model achieves maximum accuracy at 98.885%, surpassing other methods. Traditional approaches like KNN (87.98%) and random forest classifier (94.262%) were less

Tab. 2. Evaluation metrics of the proposed model.

Evaluation parameter	Value
Accuracy score	0.988
Macro averaged precision	0.979
Micro averaged precision	0.982
Macro averaged recall	0.972
Micro averaged recall	0.984
Macro averaged F1 score	0.977
Micro averaged F1 score	0.984

Tab. 3. Accuracy and loss performance of the proposed model.

Evaluation metric	Performance value
Training accuracy	0.988
Validation accuracy	0.977
Training loss	0.0318
Validation loss	0.0537

Tab. 4. Comparison of accuracy for different models in the PhishTank dataset.

Ref.	Approach	Accuracy	Dataset
[5]	Multi-model	97.81%	PhishTank
[7]	KNN	87.980%	PhishTank
[14]	NLP	97.981%	PhishTank
[15]	RFC	94.262%	PhishTank
[16]	SVM	94.130%	PhishTank
Proposed	RNN-LSTM	98.885%	PhishTank

accurate than multi-model (97.81%). With a precision of 97.981%, the NLP model is useful but not as accurate as the RNN-LSTM model. The SVM and the random forest classifier have a precision of 94.130% and 94.262%, respectively, indicating that traditional techniques are less effective than deep learning in detecting phishing URLs. The comparison indicates that the RNN-LSTM model improves phishing detection and is acceptable for accurate real-world applications.

4.2. Related Work Comparison

Table 4 compares the phishing detection methods applied to the PhishTank dataset, showcasing both traditional models like KNN, RFC, and SVM, and advanced techniques such as multimodel and NLP approaches. The proposed RNN-LSTM model achieves the highest accuracy of 98.885%, highlighting its ability to effectively leverage sequential and temporal patterns for superior phishing detection in cloud environments.

4.3. Integration in Cloud Computing Environments

The proposed phishing detection model integrates seamlessly into cloud environments, serving as a vital component of the security framework. Positioned at the network entry point, it enables real-time monitoring of incoming traffic, including URLs in emails, shared cloud storage links, and web requests, effectively preventing malicious traffic from reaching users or critical services.

Designed as a scalable microservice, the system leverages technologies like Docker and Kubernetes to dynamically adjust resources based on traffic volume. Its API-based design allows easy integration with existing security tools, such as firewalls and intrusion detection systems. Continuous learning is supported through updates from cloud-based threat intelligence feeds, ensuring adaptability to emerging phishing tactics and maintaining high detection accuracy.

Key use cases for this integration include:

- email security – analyzing and detecting phishing links embedded in emails hosted on cloud platforms,
- web gateway protection – filtering malicious URLs in real-time to block access to phishing websites,
- cloud storage monitoring – scanning shared links and files for potential phishing attempts,
- application-level protection – safeguarding cloud-based applications by monitoring API calls and user interactions for anomalies.

5. Conclusions

This paper presents a robust approach to phishing detection in cloud environments by combining RNNs with LSTM units. Leveraging sequential data processing and the ability to retain long-term dependencies, the RNN-LSTM model efficiently captured temporal patterns within cloud service interactions. This capability significantly improved accuracy and resilience in detecting phishing attempts. The model was trained on a comprehensive dataset of 10,000 cloud-based interactions, which encompassed a diverse range of applications, storage solutions, and email services. This data set facilitated robust training and ensured adaptability in various cloud scenarios.

Achieving an accuracy of 98.88%, the proposed model demonstrated the potential of deep learning to significantly enhance phishing detection in cloud computing, thereby reinforcing cybersecurity by distinguishing legitimate from malicious interactions.

Although the results are promising, there is significant potential for further research to improve and extend this work. Future research could focus on integrating advanced architectures, such as transformers or hybrid deep learning models, to enhance accuracy and efficiency. Adding real-time data streams and multilingual phishing datasets would increase the applicability and address the global nature of phishing threats. Furthermore, testing the system in real-world cloud environments would provide important insights into its scalability, reliability, and practical deployment, helping to close the gap between research and real-world applications.

References

[1] C. Sharma and C. Sharma, "Cloud Computing Security: Threats and Mitigation Strategies", *2024 International Conference on Signal Processing and Advance Research in Computing (SPARC)*, vol. 1, Lucknow, India, 2024.

[2] M. Dawood *et al.*, "Cyberattacks and Security of Cloud Computing: A Complete Guideline", *Symmetry*, vol. 15, no. 11, 2023 (https://doi.org/10.3390/sym15111981).

[3] P. Prajapati *et al.*, "Phishing E-mail Detection Using Machine Learning", *Smart Innovation, Systems and Technologies*, vol. 392, 2023 (https://doi.org/10.1007/978-981-97-3690-4_32).

[4] J.K. Samriya *et al.*, "Blockchain and Reinforcement Neural Network for Trusted Cloud Enabled IoT Network", *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2311–2322, 2024 (https://doi.org/10.1109/TCE.2023.3347690).

[5] B. Jha, M. Atre, and A. Rao, "Detecting Cloud-based Phishing Attacks by Combining Deep Learning Models", *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, Atlanta, USA, 2023 (https://doi.org/10.1109/TPS-ISA56441.2022.00026).

[6] S.R. Alotaibi *et al.*, "Explainable Artificial Intelligence in Web Phishing Classification on Secure IoT with Cloud-based Cyber-physical Systems", *Alexandria Engineering Journal*, vol. 110, pp. 490–505, 2024 (https://doi.org/10.1016/j.aej.2024.09.115).

[7] P. Ramadevi *et al.*, "Analysis of Phishing Attack in Distributed Cloud Systems Using Machine Learning", *2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, Trichirappalli, India, 2023 (https://doi.org/10.1109/ICEEICT56924.2023.10157447).

[8] U.A. Butt *et al.*, "Cloud-based Email Phishing Attack Using Machine and Deep Learning Algorithm", *Complex& Intelligent Systems*, vol. 9, pp. 3043–3070, 2023 (https://doi.org/10.1007/s40747-022-00760-3).

[9] A. Alhogail and A. Alsabih, "Applying Machine Learning and Natural Language Processing to Detect Phishing Email", *Computers and Security*, vol. 110, art. no. 102414, 2021 (https://doi.org/10.1016/j.cose.2021.102414).

[10] I. Saha *et al.*, "Phishing Attacks Detection Using Deep Learning Approach", *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2020 (https://doi.org/10.1109/ICSSIT48917.2020.9214132).

[11] M.M. Alani and H. Tawfik, "PhishNot: A Cloud-based Machine Learning Approach to Phishing URL Detection", *Computer Networks*, vol. 218, art. no. 109407, 2022 (https://doi.org/10.1016/j.comnet.2022.109407).

[12] M. Elberri, U. Tokeser, J. Rahebi, and J. Lopez-Guede, "A Cyber Defense System Against Phishing Attacks with Deep Learning Game Theory and LSTM-CNN with African Vulture Optimization Algorithm (AVOA)", *International Journal of Information Security*, vol. 23, pp. 2583–2606, 2024 (https://doi.org/10.1007/s10207-024-00851-x).

[13] E.A. Aldakheel *et al.*, "A Deep Learning-based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators", *Sensors*, vol. 23, no. 9, 2023 (https://doi.org/10.3390/s23094403).

[14] F. Sadique, R. Kaul, S. Badsha, and S. Sengupta, "An Automated Framework for Real-time Phishing URL Detection", *Proc. of the 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 335–341, 2020 (https://doi.org/10.1109/CCWC47524.2020.9031269).

[15] O. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine Learning Based Phishing Detection from URLs", *Expert Systems with Applications*, vol. 117, pp. 345–357, 2018 (https://doi.org/10.1016/j.eswa.2018.09.029).

[16] R. Rao, T. Vaishnavi, and A. Pais, "CatchPhish: Detection of Phishing Websites by Inspecting URLs", *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 813–825, 2019 (https://doi.org/10.1007/s12652-019-01311-4).

Oussama Senouci, Ph.D., Associate Professor
Computer Science Department
https://orcid.org/0000-0002-5345-0713
E-mail: oussama.senouci@univ-bba.dz
University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz

Nadjib Benaouda, Ph.D., Associate Professor
Computer Science Department
https://orcid.org/0000-0002-4361-9597
E-mail: nadjib.benaouda@univ-bba.dz
University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz

Cat Swarm Optimization with Lévy Flight for Link Load Balancing in SDN

Kwaku Kwarteng[1], Kwame O. Gyasi[1], Justice O. Agyemang[1], Kwame Agyekum[1], Kingsford Kwakye[1], Ellis M. Sani[1], Emmanuel A. Ampomah[1], and Kusi A. Bonsu[2]

[1]*Kwame Nkrumah University of Science and Technology, Kumasi, Ghana,*
[2]*Sunyani Technical University, Sunyani, Ghana*

https://doi.org/10.26636/jtit.2025.1.1773

Abstract — Efficient network communications with optimal network path selection play a key role in the modern world. Conventional path selection algorithms often face numerous challenges resulting from their limited scope of application. This research proposes a modified swarm intelligence approach, known as cat swarm optimization (CSO) with Lévy flight that is used for network link load balancing and routing optimization. CSO's quick convergence capabilities are suitable for rapid response applications; however, the approach is prone to getting stuck in local optima. Lévy flight enhances search efficiency, thus aiding in escaping local optima. CSO with Lévy flight (CSO-LF) outperforms original CSO and PSO algorithms in terms of solution quality and robustness across various benchmarks. The proposed method has been evaluated in software defined networks (SDN) with nine benchmark functions assessed. CSO-LF achieved the best scores in both the best and worst positions. When used in SDN for link load balancing, CSO-LF demonstrated lower latency and higher throughput than CSO, and lower latency and higher throughput than PSO in a fat tree topology.

Keywords — cat swarm optimization, Lévy flight, load balancing, software-defined networks

1. Introduction

Optimization plays a critical role across numerous scientific areas, as it allows to improve various algorithms of the conventional, evolutionary and nature-inspired metaheuristic variety, thus striving to solve more complex challenges. In recent years, nature-inspired algorithms have gained popularity, especially when it comes to addressing non-linear optimization tasks [1].

Software-defined networks (SDNs) allow to address some of the inherent challenges of modern network management by separating the control and data planes. This separation allows for a centralized view of the network, which facilitates a more effective implementation [2], [3]. Although SDNs alleviate some limitations in scalability and management of traditional network structures, they also introduce new challenges, especially in achieving balanced network loads. Many existing methods face difficulties in reaching a global optimum and handling non-linear dynamics (Fig. 1).

To address these issues, metaheuristic techniques, particularly those inspired by natural phenomena, such as swarm intelligence, have emerged as a promising solution. Moreover,

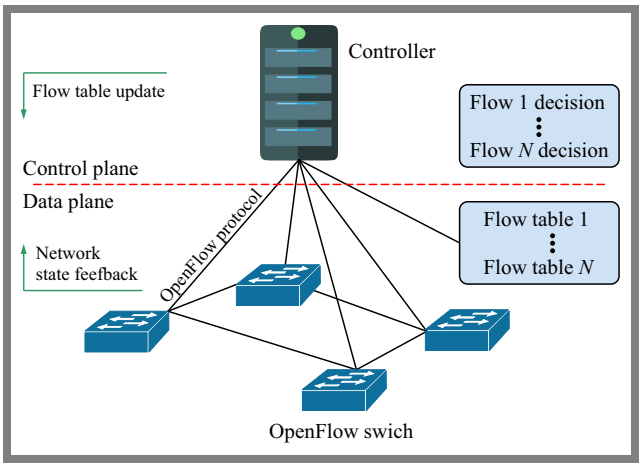

Fig. 1. Architecture of an SDN-based system.

techniques such as anomalous diffusion and Lévy flights offer the potential for improved optimization by enabling larger movements through solution spaces [1].

Bearing in mind the context defined above, this research is devoted to advancing SDN technology by utilizing cat swarm optimization with Lévy flight (CSOLF) methods. The primary objective is to improve load balancing and path optimization within SDNs. This method seeks to address the complexities of modern computer networks, promoting better scalability, efficient resource management, and improved network performance.

The contributions of this work are listed below.

- improvement of the global search capability of the original CSO algorithm by applying Lévy flight,
- implementation of the proposed algorithm in the SDN controller for link load balancing,
- comparison of throughput- and latency-related metrics of the proposed method with the original parameters of cat swarm optimization (CSO) and particle swarm optimization (PSO) algorithms.

PSO, inspired by the behavior of bird or fish swarms [4], was initially introduced as a metaheuristic algorithm for the optimization of functions. Since then, it has been applied to various optimization problems, including dynamic systems [5], pattern matching [6], traveling salesman problem [7], scheduling, and vehicle routing [8]. The CSO algorithm

based on cat behavior [9] operates in two modes, seeking and tracing, for exploration and exploitation, respectively. CSO has found applications in function optimization [10], task allocation [11], and workflow scheduling [12].

The remaining parts of this paper are structured as follows. Section 2 discusses related works. Section 3 presents the concepts of Lévy flight and CSO. Section 4 provides a comprehensive explanation and describes the implementation of the proposed algorithm. Section 5 presents the experimental analysis performed in various scenarios. Finally, concluding remarks are presented in Section 6.

2. Related Works

Swarm intelligence and other algorithms are often used in optimization and computational techniques, especially in software-defined networks (SDN) [13]. Leveraging these algorithms, SDN offers an optimal framework for scalable and programmable networks. In complex network management, these algorithms address an extensive solution space and multiple targets. Integrating heuristics with SDN improves load balancing [14]. However, studies in the literature have demonstrated that integrating Lévy flight with metaheuristic algorithms significantly enhances their efficiency, both in SDN and in other applications.

Kolodziejczyk et. al. [15] proposed particle swarm optimization (PSO) methods based on the characteristics of the Lévy distribution. This involves employing the Lévy distribution to start the swarm and using the Lévy flight as a scalar inertia coefficient. Bousmaha *et al.* [16] introduced a training technique that combines PSO with multiverse optimization using Lévy flight. This approach helps prevent premature convergence and achieves a better balance between exploration and exploitation. To address poor performance of quantum-behaved PSO (QPSO) in high-dimensional problems, paper [17] incorporated Lévy flight and straight flight (SF) strategies into QPSO. This method showed strong performance in solving engineering design optimization challenges.

Ant colony optimization (ACO) was improved in [18], based on the Lévy distribution applied to the candidate selection process, and took advantage of the Lévy flight, which increased searching speed and also ensured a better exploration of search space. In article [19], a greedy Lévy ACO was proposed, combining epsilon greedy and Lévy flight strategies to tackle complex combinatorial optimization problems. This method is developed in max-min ACO and is applied to solve the traveling salesman problem. Paper [20] introduced a hybrid max-min ant system (HMMAS) that incorporates the Lévy flight strategy to address the limitations of the traditional approach. HMMAS improves diversity by dynamically adjusting its parameters.

Verma et. al. [21] proposed modified chicken swarm optimization (MCSO) that addresses local optima and early convergence issues in basic CSO by incorporating Lévy flight as a random feature. This enables MCSO to navigate cases where conventional methods may fail to find neighboring solutions. Through experiments on various benchmark functions and pressure vessel design problems, MCSO demonstrated efficient performance, with faster convergence to global optima. Statistical analysis and comparisons with other optimization methods confirmed the effectiveness of MCSO in various problem domains. Article [22] introduced an improved artificial bee colony (ABC) algorithm incorporating Lévy flight (LABC) to improve the exploitation capability of the ABC algorithm for estimations performed in the 3-p distribution. Compared to other metaheuristic algorithms, the results demonstrated that LABC provided more precise maximum likelihood (ML) estimations. The authors of [23] developed a gray wolf optimization algorithm with dynamic adjustment of the inertia weight and a Lévy flight strategy. In the early stages of iteration, Lévy increase the probability of enhancing global search capabilities and boosts population diversity.

To recapitulate, the integration of Lévy into various metaheuristic algorithms has resulted in improvements in solving complex optimization problems. These techniques improved the balance between exploration and exploitation, improved convergence rates, and provided robust solutions to a wide class of computational challenges. However, there are still several optimization algorithms that have not yet been integrated with Lévy flight. Table 1 summarizes the findings obtained from the literature.

3. CSO and Lévy Flight

3.1. Original CSO Algorithm

CSO is a metaheuristic algorithm that mimics the behavior of cats [9]. It alternates between the seeking and tracing phases, representing local and global search for optimal solution. The cat that records the best solution will be kept in memory once the cats have been divided into these two phases, from which new positions and fitness functions will be evaluated. These processes are repeated, until the termination criteria are met [24].

In the search phase, the cat rests or observes and searches for the best solution by slightly adjusting its position and evaluating the results to avoid rapid changes, ensuring a more thorough exploration of the solution space [25].

The position of each cat is adjusted using the following equation:

$$X_{cn} = (1 \pm \text{SRD} \times R) \times X_c \,, \tag{1}$$

where X_c denotes the existing position of the cat, X_{cn} denotes the updated position of the cat, SRD denotes the search range of the selected dimension, and R denotes a random number within the range $0, \ldots, 1$. The X_{cn} parameter determines the direction of the cat's movement.

The fitness solution (FS) measure assesses the effectiveness of each candidate solution or cat in addressing the optimization problem. The FS function assigns a value that reflects how close a solution is to the optimal result, guiding the algorithm's exploration (seeking phase) and exploitation (tracing phase) behaviors. Higher fitness values lead cats to promising regions

Tab. 1. Summary of the literature review.

No.	Topic	Metrics	Solution	Limitation
[15]	PSO and Lévy flight integration	Average performance, standard deviation	Modified particle swarm optimization	Overhead
[16]	Automatic selection of hidden neurons and weights in neural networks for data classification using hybrid PSO multi-verse optimization based on Lévy flight	Standard momentum back propagation and adaptive learning rate	PSO with multi-verse optimization using Lévy flight	Scability issues
[17]	Quantum PSO with optimal guided Lévy and straight flight for solving optimization problems	Friedman rank test	PSO with straight flight and Lévy flight	High computational cost
[18]	An ant colony optimization (ACO) with Lévy flight	Throughput	Elman neural network for network optimization	Slow convergence
[19]	Improving ant colony optimization algorithm with epsilon greedy and Lévy flight	Travelling salesman and related problems	Ant colony optimization algorithm with epsilon greedy and Lévy flight	Complex
[20]	A hybrid max-min ant system by Lévy flight and opposition-based learning	Travelling salesman and related problems	Ant colony optimization with Lévy flight and opposition-based learning	Potential convergence instability
[21]	Lévy's flight guided modified chicken swarm optimization	Win-tie-loss, Bonferroni-Dunn post-hoc, and Wilcoxon tests	Modified chicken swarm optimization to solve early convergence problem of chicken swarm optimization	Single objective problem scenario
[22]	Artificial bee colony with Lévy flights for parameter estimation of 3-p Weibull distribution	Machine learning estimation tests	Artificial bee colony with Lévy flights	Limited scalability on high dimensional problems
[23]	Grey wolf optimization algorithm based on dynamically adjusting inertial weight and Lévy flight strategy	Standard test functions	Grey wolf optimization with Lévy flight	Difficulty in balancing inertia weight adjustments

of the solution space, allowing the algorithm to iteratively converge on the best possible outcome [25].

In the second tracing phase, the cat is hunting its prey in the tracing phase. The position and velocity of the prey are used by the cat to calculate its movement speed and direction after finding the prey while resting in the search phase [25]. The equation for velocity of CSO's cat k in dimension d is:

$$v_{k,d} = v_{k,d} + r_1 \times c_1(X_{\text{best},d} - X_{k,d}) \,, \tag{2}$$

where $v_{k,d}$ denotes the velocity of the cat k, $X_{best,d}$ denotes the best position of the cat, $X_{k,d}$ denotes the position of the k-th cat, c_1 is a constant and r_1 denotes a random number between 0 and 1.

With this velocity, the cat traverses the M-dimensional decision space and reports each new position. The new position is determined by the following formula:

$$X_{k,d,new} = X_{k,d,old} + v_{k,d} \,, \tag{3}$$

where $X_{k,d,new}$ represents the new position of the k-th cat, and $X_{k,d,old}$ represents the present position of the k-th cat.

Completion of the algorithm is determined based on the achievement of termination conditions [26] which include the number of iterations, progress, and time.

3.2. Lévy Flight

Lévy flight is a class of non-Gaussian random walks whose step length is drawn from the Lévy distribution, often in terms of a simple power law equation [27]. The simple power law equation is given as:

$$L \sim |s|^{-\mu}, \quad 0 < \mu \leqslant 2 \,, \tag{4}$$

where L is the length of the step, s is the step and μ is the Lévy exponent, controlling the scale of the steps.

In practice, a Lévy flight step can be modeled as:

$$L = \frac{\text{rand}()}{|\,\text{rand}()\,|^{\frac{1}{\mu}}} \,,$$

where $\text{rand}()$ generates normally distributed random numbers and μ controls the step size.

Tab. 2. Group of test functions used in numerical evaluations [29].

Function	Expression	Dim	Search range
Ackley	$f_1 = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + \exp(1)$	2	$-5 \leqslant x \leqslant 5$
Drop wave	$f_2 = -\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{0.5(x_1^2+x_2^2)+2}$	2	$-5.12 \leqslant x \leqslant 5.12$
Bukin	$f_3 = 100\sqrt{\lvert x_2 - 0.01x_1^2\rvert} + 0.01\lvert x_1 + 10\rvert$	2	$-15 \leqslant x_1 \leqslant -5$, $-3 \leqslant x_2 \leqslant 3$
Three hump camel	$f_4 = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	2	$-5 \leqslant x \leqslant 5$
Matyas	$f_5 = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	$-10 \leqslant x \leqslant 10$
Bohachevsky	$f_6 = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	5	$-100 \leqslant x \leqslant 100$
Sphere	$f_7 = \sum_{i=1}^{d} x_i^2$	2	$-5.12 \leqslant x \leqslant 5.12$
Sum squares	$f_8 = \sum_{i=1}^{d} ix_i^2$	2	$-10 \leqslant x \leqslant 10$
Sum of different powers	$f_9 = \sum_{i=1}^{d} \lvert x_i\rvert^{i+1}$	2	$-1 \leqslant x \leqslant 1$

This equation models the mixture of small, local searches (short steps) and long-range exploration (large jumps) characteristic of Lévy flight, making it useful for global optimization in metaheuristics. Lévy flights are more efficient than Brownian random walks when it comes to exploring large-scale search spaces. There are many reasons to explain this efficiency, one of which stems from the fact that the variance of Lévy flights increases much faster than the linear relationship of Brownian random walks [28].

4. Proposed Method

Despite the many CSO variants available in the literature, the problems of premature convergence and generating inefficient results continue to persist. The Lévy flight method is used to solve these problems and enables CSO to generate more efficient results. This method ensures that the CSO, which is unable to perform the global search well, will enjoy better search efficiency and will not be trapped in local minima. In the CSO-LF method, the Gaussian random walks used in the tracing mode are replaced with Lévy flight.

4.1. CSO-LF Algorithm

The CSO-LF follows a process similar to the original CSO in the seeking phase. However, the movement that occurs in the tracing phase is where the novelty of the algorithm is shown.

In the tracing phase, the new position of the cat is calculated by:

$$X_{k,d,\text{new}} = \left[\alpha \cdot L(\beta)\right] X_{k,d,\text{old}} + v_{k,d}\ , \tag{5}$$

where α is the scaling parameter that determines the step size in the Lévy flight and $L(\beta)$ denotes a random number vector obtained using Lévy distribution with the β exponent.

In Eq. (5), the movement taking place in the tracing phase, which is originally powered by the Brownian random walk, is replaced with the Lévy flight. In the Lévy flight, the step size follows a probability distribution, which allows for occasional long jumps [27]. The scaling parameter affects the frequency with which longer jumps occur, and thus, it is a very important parameter in ensuring the overall performance of the algorithm. In this paper, the scaling parameter has been fine-tuned to ensure optimal search efficiency.

The algorithm is terminated based on achieving the termination conditions. In this study, is defined by a predetermined

number of iterations. The algorithm continues until the last iteration has been completed. The pseudocode of the proposed solution is given as Algorithm 1.

4.2. Test Functions

A selection of unimodal and multimodal test functions [29] was used to perform a numerical evaluation for the suggested approach (as shown in Tab. 2). The functions were chosen for their diverse shapes and scaling ability. f_1, f_2 and f_3 are functions with many local minima, f_4 is a valley-shaped function, f_5 is a plate-shaped function and f_6, f_7, f_8 and f_9 are bowl-shaped functions.

The initial population was generated from a uniform distribution across the search space, proportional to its specified boundaries. Table 2 provides the range limits for all functions. The maximum iteration count was set to 100 for a population size of 50.

The numerical evaluations were carried out in five trials, with each trial producing a set of optimization results. For each function, three key metrics were recorded: the best, the worst and the average best solutions in the five trials. Here, the "worst" outcome represents the highest (least optimal) function value obtained in any of the five tests, indicating the algorithm's most unfavorable performance. This metric provides information on variability and resilience under less favorable conditions. These metrics were then ranked using a standard competition ranking system, where lower scores in the specific categories indicate more effective algorithms.

Algorithm 1 Pseudo-code for cat swarm optimization with Lévy flight (CSO-LF)

Initialize parameters
α = scaling parameter (step size in Lévy flight)
β = Lévy distribution exponent
Number of cats (agents), iterations, dimensions
Initialize positions $X[k][d]$ and velocities $v[k][d]$ for each cat
while stopping condition is not met **do**
    for each cat k **do**
        if in the seek phase **then**
            Perform the seek phase
            (according to the standard CSO)
        else in the tracing phase
            for each dimension d of the cat k **do**
                Generate a random vector $L(\beta)$
                Compute a new position:
                $X[k][d]_{\text{new}} = \big(\alpha \cdot L(\beta)\big) \cdot X[k][d]_{\text{old}} + v[k][d]$
            end for
            Update the cat position $X[k]$ with $X[k]_{\text{new}}$
        end if
    end for
    Update velocities $v[k][d]$ based on the new position
    Evaluate the fitness of each cat's new position
    Identify and update the best position found so far
end while
Output the best position found and its fitness value

Tab. 3. Notations used.

Notation	Description
G	Topology of the given network
V	Set of all switches
E	Set of all links that connect the switches together
$\phi_{s,d}$	Set of all feasible paths for switch pair v_s and v_d
I	Initial population
$F(k)$	Fitness function of the k-th cat
$P_{utilization_k}$	Time it takes for a packet to be transmitted on the k-th path
$P_{congestion_k}$	Available bandwidth on the k-th path

4.3. Network Modelling

The proposed CSO-LF is modeled as a load balancing problem that was equated to find the path with the least cost in the SDN environment whose parameters are shown in Tab. 3. From the data plane perspective, the topology of the network can be modeled by:

$$G = (V, E)\,, \tag{6}$$

where V is a set of switches, given by $V = v_1, \ldots, v_n$ with $n = |N|$ and E is a set of links that connect the switches to each other, given by $E = e_1, \ldots, e_m$ with $m = |E|$.
The algorithm seeks to minimize path utilization $P_{utilization}$ and path congestion $P_{congestion}$ of a given topology. Based on the condition above, the objective function is designed, namely:

$$\text{Fitness function} = \min(P_{utilization} \times P_{congestion})\,, \tag{7}$$

In CSO-LF, the SDN controller first generates, randomly, the initial population, i.e. a set of cats. The position of each cat, which is a possible and available path, consists of a set of switches that can connect the sender side and the receiver side. This is defined by the following:

$$C = \big\{c_k \mid c_k \in \phi_{(s,d)}, (s,d) \in V\big\},\ \forall k, \quad k = 1, \ldots, K, \tag{8}$$

where $\phi_{(s,d)}$ is a set of all feasible paths for each pair v_s and v_d.

Tab. 4. Network topologies used in evaluation.

Mesh topology	Fat tree topology
10 switches, 10 hosts	20 switches, 16 hosts
20 switches, 20 hosts	45 switches, 54 hosts
30 switches, 30 hosts	80 switches, 124 hosts
40 switches, 40 hosts	125 switches, 250 hosts
50 switches, 50 hosts	180 switches, 432 hosts

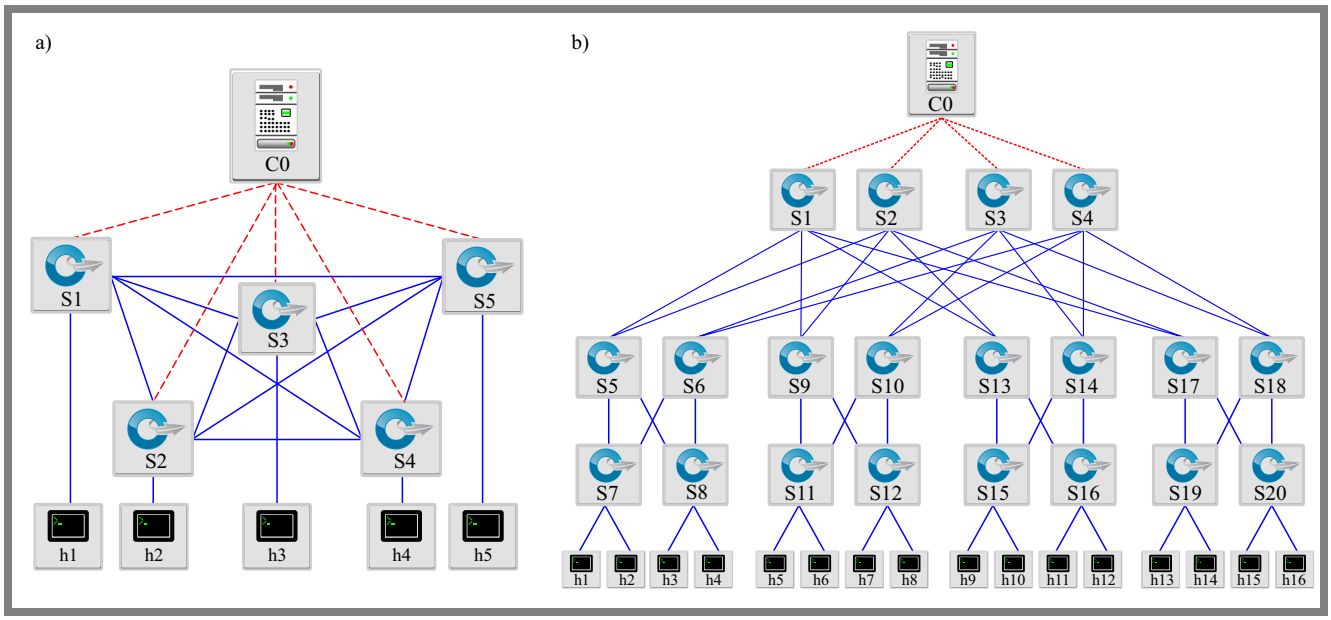

Fig. 2. Test topologies: a) mesh and b) fat tree.

Each path is made up of switches V_k and links E_k, where $V_k = \{v_i | v_i \in c_k\}$ is the set of switches in cluster c_k and $E_k = \{e_{i,j} | e_{i,j} \in c_k\}$ is the set of links in cluster c_k.

After initialization, the fitness value is calculated for each initialization condition. In this paper, a fitness function corresponding to path utilization and congestion is taken into consideration.

Let us assume that the generated initial population is I, such that $I = p_1, p_2, \ldots, p_k$, the fitness function of cat k is given as:

$$F(k) = \min \left(P_{utilization_k} \times P_{congestion_k},\ k \in C \right), \quad (9)$$

where $P_{utilization_k}$ defines how long a packet awaits to be transmitted on the k-th path denoted by:

$$P_{utilization_k} = \frac{\lambda_k}{\mu_k}, \quad (10)$$

This provides a good cost metric, since it is low for low loads and goes to infinity for very high loads. $P_{congestion_k}$ defines the available bandwidth of the path and is inversely proportional to its bandwidth, denoted by:

$$P_{congestion_k} = \frac{1}{P_{bandwidth_k}}. \quad (11)$$

The CSO-LF algorithm uses a swarm of cats to perform a search for the possible path, with each cat representing a candidate path. The algorithm iteratively updates the position of each cat based on a Lévy flight step. The new position is chosen from the list of possible paths based on the fitness of the new position. If the fitness of the new position is greater than the fitness of the current position, the cat moves to the new location. This process continues until the assumed number of iterations is reached.

4.4. Implementing CSO-LF Generated Paths in SDN

In SDN environments, after the CSO-LF algorithm has determined the best path for network traffic, the selected path must be implemented as flow entries in the switches controlled by the SDN controller. This process involves several steps:

- **Path encoding and flow rule generation**. The path selected by the CSO-LF algorithm needs to be encoded into a series of flow rules. Each flow rule specifies how the network traffic that satisfies certain criteria should be treated. Typically, these flow rules include such information as source and destination addresses, ports, and quality-of-service requirements.
- **Flow table update**. The SDN controller processes the received flow rules and updates the flow tables of the relevant switches. Flow tables are used by switches to determine how to forward network traffic. The SDN controller installs the new flow entries in the switches along the selected path. If necessary, it may also remove any old flow entries associated with the previous path.
- **Flow table matching**. When network traffic arrives at a switch, the switch examines its flow table to determine how to handle the traffic. The switch matches the incoming packets with the installed flow rules. If a match is found, the switch takes the specified action, such as forwarding the traffic along the path defined by the flow rule.
- **Packet forwarding**. In the next step, network traffic is forwarded by the switches along the path defined by the flow rules. Switches ensure that traffic follows the chosen path, and can also perform such actions as quality of service (QoS) shaping, security checks, and other functions specified by the flow rules.
- **Dynamic path adaptation**. In SDN environments, network conditions and requirements may change. Therefore, the SDN controller continuously monitors the network and can

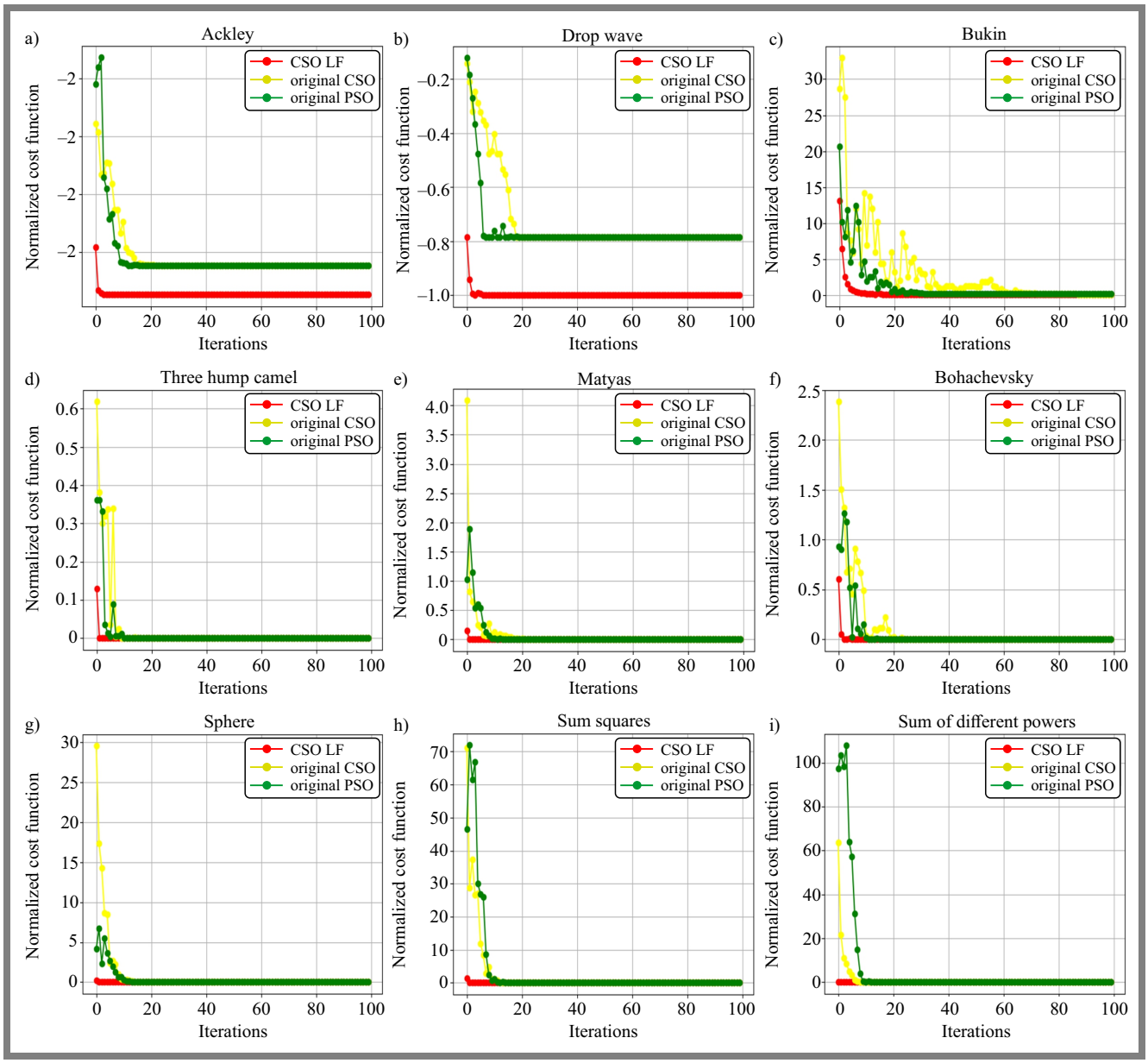

Fig. 3. Convergence plots for the three algorithms for an initial population of 50 agents in 100 iterations.

dynamically adapt flow entries if needed. If the controller detects changes in the network, it can recalculate paths (using the CSO-LF algorithm), generate new flow rules, and update switches to reflect these changes.

By following these steps, the selected path generated by the CSO-LF algorithm is implemented in the SDN network. This dynamic and software-driven approach to path selection and flow rule management is one of the key advantages of SDN, as it facilitates efficient traffic engineering and offers adaptability in response to changing network conditions.

4.5. Configuration and Setup

The software environment used for the CSO-LF experiment is the Python 3.8.10 programming language. Other parameter settings of the algorithm include: SMP = 2, CDC = 1, SRD = 0.1, SPC = false, MR = 0.67. The parameters of the Lévy component include $\beta = 1.5$ and $\alpha = 0.2$. SDN simulations were conducted in the Oracle VM VirtualBox 6.1 hypervisor with a Linux Ubuntu (64-bit) operating system, Mininet emulator, RYU controller, and an OpenFlow communication protocol.

Two different network topologies of different sizes were used to test the proposed method: mesh and fat tree topology. These topologies were chosen for their different behaviors and logical arrangements.

The mesh topology provides high redundancy and fault tolerance, while the tree topology offers hierarchical organizational design, fault tolerance, and high-level scalability.

Figure 2 shows the different topologies connected to a centralized controller, while Tab. 4 shows different topology sizes used to test the proposed algorithm.

4.6. Performance Evaluation Method

In this research, latency and network throughput are considered to be key performance metrics. Throughput represents the rate at which data is successfully transferred from a source to a destination over a communication channel [28]. The equation to calculate throughput is:

$$\text{Throughput} = \frac{\text{Amount of data transferred}}{\text{Time taken}} . \tag{12}$$

Latency, also known as delay, refers to the time it takes a data packet to travel from its source to its destination. It includes various components, such as transmission delay, propagation delay, queueing delay, and processing delay. The equation for calculating latency depends on the specific components involved in the process, but may be simplified as:

$$\begin{aligned}\text{Latency} = {} & \text{Transmission delay} + \text{Propagation delay} \\ & + \text{Queuing delay} + \text{Processing delay}\end{aligned} . \tag{13}$$

5. Results and Discussions

In this section, the experimental results of the proposed methodology are analyzed and its performance is evaluated.

5.1. CSO-LF

Three algorithms (i.e., CSO, PSO, and CSO-LF) are taken through the optimization process for the chosen test functions. In Tab. 5, the results of numerical experiments are displayed. The results indicate that CSO-LF consistently achieves the highest precision across all the test functions, especially in Ackley, three-hump camel, Bohachevsky, sphere, sum squares, and sum of different powers. It excels in locating positions closest to the global minima with low scores, showcasing superior convergence. In general, CSO-LF emerges as the most accurate algorithm. This comparison suggests that CSO-LF is the most effective solution for optimization tasks that require a high degree of accuracy.

Figure 3 presents a collection of graphs comparing the convergence properties of all three algorithms. The plots depict the best optimization outcome after five trials.

The proposed approach, which converges within a comparatively smaller number of iterations, outperforms the other two algorithms (CSO and PSO), according to the convergence plots.

5.2. CSO-LF in SDN

The performance of CSO-LF in SDN is evaluated using latency and throughput metrics across two different topologies. The best-performing algorithm should have the lowest latency and the highest throughput. In Fig. 4a, it was observed that CSO-LF generally underperformed in terms of latency, but Fig. 4b shows it outperformed the original CSO as far as the throughput measure is concerned.

In Figs. 4c–d, one may see that CSO-LF outperformed CSO in terms of latency and throughput, while Fig. 4a shows that the proposed method had the highest latency. This is because

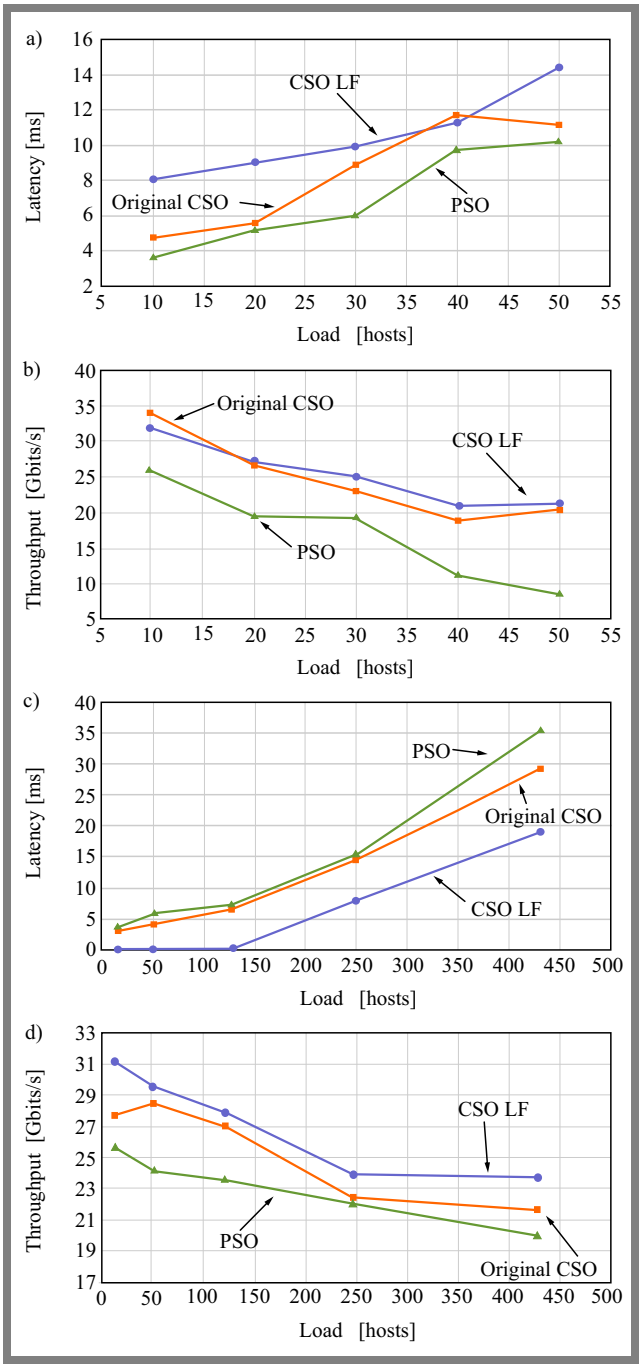

Fig. 4. Simulated latency and throughput results of the three algorithms for: a-b) mesh topology and c-d) fat tree topology.

in a small-scale network, the distances between nodes are relatively short. However, Lévy flights have occasional long jumps, which may cause a search to go beyond adjacent nodes and take a considerable amount of time to locate the optimal path for transmission of a packet – hence the higher latency. Furthermore, PSO recorded the lowest latency, highlighting its potential use case in relatively smaller networks.

From Fig. 4b, it was also observed that the proposed method had a lower throughput varying between 10 and 20 hosts, but its value became higher as the number of hosts increased. Again, Lévy flight steps take more time to converge to an optimal solution in smaller networks due to the initial explo-

Tab. 5. Comparison between CSO-LF, CSO, and PSO based on their positions and scores.

Function	Algorithm	Best score	Best position	Worst score	Worst position
Ackley	CSO-LF	–9.46	[2.6E–08, 6.9E–13]	3.58	[–8.19, –3.67]
	CSO	–8.46	[6.28, 6.28]	3.49	[–7.52, –2.99]
	PSO	–8.46	[6.28, 6.28]	3.31	[–3.96, 8.13]
Drop wave	CSO-LF	–1.00	[–7.07E–15, –9.58E–13]	–0.01	[–0.25, 0.66]
	CSO	–0.79	[0.18, 0.67]	–0.05	[0.60, 4.74]
	PSO	–0.79	[–0.09, 0.06]	–0.01	[3.73, –7.32]
Bukin	CSO-LF	0.10	[6.43E–19, –3.45E–20]	64.00	[–0.11, 0.41]
	CSO	0.24	[9.99, 0.99]	235.6	[7.66, –4.96]
	PSO	0.20	[10.00, 1.00]	46.98	[15.60, 2.21]
Three hump camel	CSO-LF	7.76E–34	[–2.76E–19, –2.78E–17]	140 779	[–9.83, 5.79]
	CSO	9.30E–08	[1.02E–04, –3.24E–04]	148 414	[9.92, –1.84]
	PSO	1.06E–25	[2.47E–13, –1.32E–13]	18 325	[7.07, –3.21]
Matyas	CSO-LF	1.06E–31	[–2.36E–21, 6.38E–16]	0.22	[–0.09, –0.05]
	CSO	4.27E–05	[–0.01, –0.01]	123.87	[8.68, –9.85]
	PSO	3.83E–18	[4.03E–09, 2.73E–09]	3.26	[–4.62, –2.87]
Bohachevsky	CSO-LF	0.0	[–2.80E–27, –2.05E–28]	0.60	[–0.56, –0.07]
	CSO	1.23E–10	[2.23E–06, –1.24E–06]	171.16	[7.65, 7.48]
	PSO	0.0	[–1.36E–09, 1.23E–09]	59.59	[2.19, –5.20]
Sphere	CSO-LF	1.11E–24	[1.57E–14, –1.05E–12]	2.36	[0.93, –0.61]
	CSO	7.73E–06	[–0.01, 0.01]	121.58	[4.77, –5.93]
	PSO	2.23E–17	[2.06E–09, 1.99E–10]	82.70	[3.04, 1.69]
Sum squares	CSO-LF	1.69E–20	[4.33E–11, 1.29E–15]	976.26	[–1.72, 8.39]
	CSO	2.71E–06	[–0.01, 0.01]	703.72	[1.32, 7.91]
	PSO	1.42E–17	[–2.87E–10, 5.55E–10]	323.38	[0.19, 1.11]

Tab. 6. Average performance for mesh and fat tree topologies.

Topology	Latency [ms]			Throughput [Gbps]		
	CSO-LF	CSO	PSO	CSO-LF	CSO	PSO
Mesh	9.50	8.37	6.82	25.26	22.50	16.85
Fat tree	5.48	11.51	13.16	27.27	23.40	23.02

Tab. 7. Percentage difference in performance between CSO-LF and CSO.

Topology	Latency	Throughput
Mesh	–32.84%	39.94%
Fat tree	82.40%	16.90%

ration phase, which can result in lower throughput during this convergence period. In larger networks, the algorithm has more opportunities to explore and discover better paths, and the convergence phase may be shorter relative to the size of the network, leading to larger throughput values.

Tab. 8. Percentage difference in performance between CSO-LF and PSO.

Topology	Latency	Throughput
Mesh	–12.65%	11.56%
Fat tree	70.98%	15.28%

Tables 7–8 show the percentage differences in performance between CSO-LF and two other optimization algorithms, CSO and PSO, across two network topologies.

In the mesh topology, CSO-LF's latency is 12.65% higher compared to CSO, indicating that CSO performs better in terms of latency in this topology. In terms of throughput, CSO-LF shows an 11.56% improvement over CSO, suggesting that CSO-LF achieves higher throughput in the mesh topology. In the fat tree topology, the percentage differences are more substantial. CSO-LF exhibits a significantly lower latency (70.98%) compared to CSO, indicating a substantial improvement in latency performance. For throughput,

CSO-LF also shows a 15.28% improvement over CSO in the fat tree topology, although the difference is not as significant as in the case of latency.

In the mesh topology, CSO-LF shows a latency that is 32.84% higher compared to PSO, indicating that PSO performs better in terms of latency in this network topology. However, in terms of throughput, CSO-LF demonstrates a substantial improvement (39.94%) over PSO, suggesting that CSO-LF achieves significantly higher throughput in the mesh topology compared to PSO. In the fat tree topology, CSO-LF again exhibits a significant improvement over PSO, with a difference of 82.40% in terms latency and 16.90% in terms of throughput. This indicates that CSO-LF outperforms PSO by a large margin, both in terms of latency and throughput, in the fat tree topology.

In general, the latency increased and the throughput decreased with increasing network size. However, CSO-LF generally shows superior performance compared to CSO and PSO in terms of both latency and throughput. The analysis reveals that the proposed method exhibits lower latency, resulting in faster response and reduced data transmission delays. Additionally, it achieves higher throughput, enabling more efficient data transfer and improved network capacity utilization.

6. Conclusions

This research introduces CSO-LF as a prospective solution for tackling optimization problems, particularly in SDN link load balancing. While CSO demonstrates rapid convergence, making it ideal for applications requiring quick responses, its vulnerability to become stuck in local optima poses a limitation. CSO-LF addresses this issue by integrating the Lévy flight technique, thus augmenting search optimality and offering improved performance in navigating complex optimization landscapes.

The proposed method was evaluated on nine popular functions. The numerical results showed that CSO-LF achieved the best scores in terms of the best and worst positions. When implemented in SDN for link load balancing, CSO-LF recorded a lower latency and a throughput that was 15.28% higher compared to CSO, and latency that was 82.40% lower when compared to PSO in the fat tree topology. Its versatility suggests potential applications in controller placement, virtual network mapping, flow entry optimization, and signal processing.

Future research efforts should consider investigating the scaling parameter, which plays a pivotal role in determining Lévy flight step sizes. Furthermore, CSO-LF's application to solve various networking optimization challenges beyond load balancing should be explored as well, and comparative studies with other algorithms should be conducted for broader validation. Evaluation of its scalability and performance in even larger and more complex network environments (such as multicontroller scenarios) is crucial for real-world use.

References

[1] *Nature-Inspired Algorithms and Applied Optimization*, ed. by X.-S. Yang, Springer, Cham, 341 p., 2018 (https://doi.org/10.1007/978-3-319-67669-2).

[2] P. Goransson, C. Black, and T. Culver, *Software Defined Networks, A Comprehensive Approach*, 2nd ed., Elsevier, Cambridge, 2017 (ISBN: 9780128045794).

[3] H. Qi and K. Li, *Software Defined Networking Applications in Distributed Datacenters*, Springer, Cham, 76 p., 2016 (https://doi.org/10.1007/978-3-319-33135-5).

[4] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization", *Proc. of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995 (https://doi.org/10.1109/ICNN.1995.488968).

[5] X. Hu and R.C. Eberhart, "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems", *Proc. of the 2002 Congress on Evolutionary Computation. CEC'02*, pp. 1666–1670, 2002 (https://doi.org/10.1109/CEC.2002.1004492).

[6] M. Kong, P. Tian, and Y. Kao, "A New Ant Colony Optimization Algorithm for the Multidimensional Knapsack Problem", *Computers & Operations Research*, vol. 35, pp. 2672–2863, 2008 (https://doi.org/10.1016/j.cor.2006.12.029).

[7] V. Pureza and P.M. Franca, "Vehicle Routing Problems via Tabu Search Metaheuristic", Centre De Recherche Sur Les Transports Publication, pp. 142–149, 1991.

[8] H. Keller, U. Pferschy, and D. Pisinger, *Knapsack Problem*, Springer, Berlin, 568 p., 2003 (https://doi.org/10.1007/978-3-540-24777-7).

[9] S.C. Chu, P.W. Tsai, and J.S. Pan, "Cat Swarm Optimization", *PRICAI 2006: Trends in Artificial Intelligence*, pp. 854–858, 2006 (https://doi.org/10.1007/978-3-540-36668-3_94).

[10] O. Bozorg-Haddad, *Advanced Optimization by Nature-Inspired Algorithms*, Springer, Singapore, 174 p., 2018 (https://doi.org/10.1007/978-981-10-5221-7).

[11] I. Boussaid, J. Lepagnot, and P. Siarry, "A Survey on Optimization Metaheuristics", *Information Sciences*, vol. 237, pp. 82–117, 2013 (https://doi.org/10.1016/j.ins.2013.02.041).

[12] M. Andresen *et al.*, "Simulated Annealing and Genetic Algorithms for Minimizing Mean Flow Time in an Open Shop", *Mathematical and Computer Modelling*, vol. 48, pp. 1279–1293, 2008 (https://doi.org/10.1016/j.mcm.2008.01.002).

[13] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks", *ACM SIGCOMM Computer Communication Review*, vol. 44, pp. 87–98, 2014 (https://doi.org/10.1145/2602204.2602219).

[14] M. Hamdan *et al.*, "A Comprehensive Survey of Load Balancing Techniques on Software-defined Network", *Journal of Network and Computer Applications*, vol. 174, 2021 (https://doi.org/10.1016/j.jnca.2020.102856).

[15] J. Kolodziejczyk and Y. Tarasenko, "Particle Swarm Optimization and Lévy Flight Integration", *Procedia Computer Science*, vol. 192, pp. 4658–4671, 2021 (https://doi.org/10.1016/j.procs.2021.09.244).

[16] R. Bousmaha, R.M. Hamou, and A. Amine, "Automatic Selection of Hidden Neurons and Weights in Neural Networks for Data Classification Using Hybrid Particle Swarm Optimization, Multi-verse Optimization Based on Lévy Flight", *Evolutionary Intelligence*, vol. 15, pp. 1695–1714, 2022 (https://doi.org/10.1007/s12065021-00579-w).

[17] X. Liu, G.-G. Wang, and L. Wang, "LSFQPSO: Quantum Particle Swarm Optimization with Optimal Guided Lévy Flight and Straight Flight for Solving Optimization Problems", *Engineering with Computers*, vol. 38, pp. 4651–4682, 2022 (https://doi.org/10.1007/s00366-021-01497-2).

[18] Y. Liu and B. Cao, "A Novel Ant Colony Optimization Algorithm with Lévy Flight", *IEEE Access*, vol. 8, pp. 67205–67213, 2020 (https://doi.org/10.1109/ACCESS.2020.2985498).

[19] Y. Liu, B. Cao, and H. Li, "Improving Ant Colony Optimization Algorithm with Epsilon Greedy and Lévy Flight", Complex and

Intelligent Systems, vol. 7, pp. 1711–1722, 2021 (https://doi.org/10.1007/s40747-020-00138-3).

[20] Z. Zhang, Z. Xu, S. Luan, and X. Li, "A Hybrid Max-min Ant System by Lévy Flight and Opposition-based Learning", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 35, 2021 (https://doi.org/10.1142/S0218001421510137).

[21] S. Verma, S.P. Sahu, and T.P. Sahu, "MCSO: Lévy's Flight Guided Modified Chicken Swarm Optimization", *IETE Journal of Research*, vol. 70, 2024 (https://doi.org/10.1080/03772063.2023.2194265).

[22] A. Yonar and N.Y. Pehlivan, "Artificial Bee Colony with Lévy Flights for Parameter Estimation of 3-p Weibull Distribution", *Iranian Journal of Science and Technology*, vol. 44, pp. 851–864, 2020 (https://doi.org/10.1007/s40995-020-00886-4).

[23] Y. Chen, J. Xi, H. Wang, and X. Liu, "Grey Wolf Optimization Algorithm Based on Dynamically Adjusting Inertial Weight and Lévy Flight Strategy", *Evolutionary Intelligence*, vol. 16, pp. 917–927, 2023 (https://doi.org/10.1007/s12065-022-00705-2).

[24] X.-S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier, Waltham, 222 p., 2014 (https://doi.org/10.1016/C2013-0-01368-0).

[25] X.-S. Yang, *Nature-inspired Algorithms and Applied Optimization*, Springer, Cham, 341 p., 2018 (https://doi.org/10.1007/978-3-319-67669-2).

[26] A.O. Jefia, S.I. Popoola, and A.A. Atayero, "Software-defined Networking: Current Trends, Challenges, and Future Directions", *Proc. of the International Conference on Industrial Engineering and Operations Management*, pp. 1677–1685, 2018 (https://doi.org/10.46254/NA03.20180435).

[27] A. Chechkin, R. Metzler, J. Klafter, and V.Y. Gonchar, "Introduction to the Theory of Lévy Flights", in: *Anomalous Transport: Foundations and Applications*, pp. 129–162, 2008 (https://doi.org/10.1002/9783527622979.ch5).

[28] A.A. Al-Temeemy, J.W. Spencer, and J.F. Ralph, "Lévy Flights for Improved Ladar Scanning", *2010 IEEE International Conference on Imaging Systems and Techniques*, Thessaloniki, Greece, 2010 (https://doi.org/10.1109/IST.2010.5548519).

[29] S. Surjanovic and D. Bingham, "Optimization Test Problems", *Virtual Library of Simulation Experiments: Test Functions and Datasets*, SFU [Online] (https://www.sfu.ca/~ssurjano/optimization.html).

Kwaku Kwarteng, M.Phil.
Department of Telecommunication Engineering
https://orcid.org/0009-0009-4416-8335
E-mail: akyeampongkwaku30@gmail.com
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
https://www.knust.edu.gh

Kwame O. Gyasi, Ph.D.
Department of Telecommunication Engineering
https://orcid.org/0000-0002-4923-4452
E-mail: kotenggyasi@knust.edu.gh
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
https://www.knust.edu.gh

Justice O. Agyemang, Ph.D.
Department of Telecommunication Engineering
https://orcid.org/0000-0002-9949-3823
E-mail: justice.agyemang@knust.edu.gh
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
https://www.knust.edu.gh

Kwame Agyekum, Ph.D.
Department of Telecommunication Engineering
https://orcid.org/0000-0002-7935-9950
E-mail: kooagyekum@knust.edu.gh
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
https://www.knust.edu.gh

Kingsford Kwakye, Ph.D.
Department of Telecommunication Engineering
https://orcid.org/0009-0006-0900-1685
E-mail: ksobengkwakye@knust.edu.gh
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
https://www.knust.edu.gh

Ellis M. Sani, Ph.D.
Department of Telecommunication Engineering
https://orcid.org/0000-0001-9344-2075
E-mail: smellis.coe@knust.edu.gh
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
https://www.knust.edu.gh

Emmanuel A. Ampomah, Ph.D.
Department of Telecommunication Engineering
https://orcid.org/0000-0001-6498-4000
E-mail: eaffume@gmail.com
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
https://www.knust.edu.gh

Kusi A. Bonsu, Ph.D.
Department of Electrical & Electronic Engineering
https://orcid.org/0000-0002-5474-5206
E-mail: kusiankrah@stu.edu.gh
Sunyani Technical University, Sunyani, Ghana
https://stu.edu.gh

FPGA-based Low Latency Square Root CORDIC Algorithm

Mariusz Węgrzyn[1], Stepan Voytusik[2], and Nataliia Gavkalova[3]

[1]*Cracow University of Technology, Cracow, Poland,*
[2]*Lviv Polytechnic National University, Lviv, Ukraine,*
[3]*Warsaw University of Technology, Warsaw, Poland*

https://doi.org/10.26636/jtit.2025.1.1950

Abstract — The coordinate rotation digital computer (CORDIC) algorithm is a popular method used in many fields of science and technology. Unfortunately, it is a time-consuming process for central processing units (CPUs) and graphics processing units (GPUs), and even for specialized digital signal processing (DSP) solutions. The CORDIC algorithm is an alternative for Newton-Raphson numerical calculation and for the FPGA based resource-expensive look-up-table (LUT) method. Various modifications of the CORDIC algorithm allow to speed up the operation of hardware in edge computing devices. With that context taken into consideration, this article presents a fast and accurate square root floating point (SQRT FP) CORDIC function which can be implemented in field programmable gate arrays (FPGAs). The proposed algorithm offers low-complexity, decent accuracy and speed, and is sufficient for digital signal processing (DSP) applications, such as digital filters, accelerators for neural networks, machine learning and computer vision applications, and intelligent robotic systems.

Keywords — computer vision, CORDIC algorithm, FPGA, numerical methods, reconfigurable computing systems

1. Introduction

The current methods by means of which the square root (SQRT) calculation approach is implemented in hardware continue to suffer from numerous drawbacks that limit their practical use. Software developers and researchers of many real-time DSP applications face challenges related to computational accuracy and speed [1], as well as optimization of hardware resources required to run square root algorithms [2]. In the Newton-Raphson numerical method that is commonly used for computing the square root, the precision level depends on the initial guess and requires significant computational resources, due to its reliance on iterative multiplication [3], [4].

Alternative multiplicative methods have a quadratic type of convergence, and thus may speed up the computation process. These methods perform a number of iterations of a fused multiply-add (FMA) operation, with the latency of a single FMA being in the range of 3 and 6 cycles [5]. For a low-precision SQRT computation, look-up table or low-degree polynomial approximation methods can be applied [1]. However, high demand for FPGA resources is an additional disadvantage here. This problem has been partially solved by iterative or digit-recurrence methods presented in [6], [7], which are characterized by linear convergence.

In order to overcome the abovementioned issues, the coordinate rotation digital computer (CORDIC) algorithm was proposed [8]–[10].

The main drawback of the CORDIC method [11], [12] is its low speed resulting from the use of linear convergence, where only one correct bit of the result per iteration is given. The number of iterations depends on the precision of the required data. Therefore, latency is a main research problem, especially when the algorithm operates on large bit-width vectors [13].

To simplify hardware implementation, a CORDIC method with angle recoding has been proposed in [7], [10], reducing computational complexity to two iteration equations only. Other articles focus on improving the speed of the method, i.e. reducing the number of iterations, proposing hybrid structures that use, sequentially, three different methods: table-based + CORDIC + piece-wise linear multiplication (linear approximation) [5], [14]–[17].

Another proposal to accelerate the CORDIC algorithm consisted in the introduction of the pre-rotation technique [13]. Moreover, this algorithm can be executed in various forms: classical [12], using higher-order iterative formulas [16], [18], without recoding [13], and with angle recoding [19], [20]. The simplest solution – in terms of hardware implementation – is CORDIC with angle recoding [10], [18]. However, a significant drawback lies in the large amounts of memory (LUT type) required for large values of m (a table of size not less than $2^{\frac{m}{3}} \times m$ bits is needed). Furthermore, the output multipliers are implemented in the {-1, 1} basis, preventing the use of multipliers that are part of the DSP blocks in modern FPGA devices.

The CORDIC algorithm is widely applied to calculate various mathematical functions, including SQRT [4], [10], [21]. An example of the application of the new numerical method for SQRT calculations is presented in [22], where an efficient design of a Kalman filter is implemented. The authors plan to apply the CORDIC algorithm to improve digital filters for telecommunications applications. Thus, we have the intention of developing further effective methods for the calculation of trigonometric functions using the CORDIC style. The said algorithm can also be relied upon to calculate nonlinear

functions, such as $\mathrm{th}(x)$, which are useful for implementing neural networks. Its other applications include low-resource microprocessors without the floating point unit (FPU), as our algorithm converts from floats to integers and all calculations are executed on integer-type numbers, with the result them being converted back to the floating point type.

The goal of this work is to propose a new efficient SQRT floating point type CORDIC algorithm with low latency and moderate FPGA resource requirements. Our proposal utilizes an original methodology of converting floating numbers to integers and vice versa, in order to optimize the use of FPGA hardware resources and to improve calculation efficiency.

The article is organized as follows. Section 2 reviews various methodologies and hardware implementations of square root algorithms. Section 3 introduces the proposed methodology. Section 4 introduces the proposed algorithm and describes the details of the FPGA implementation, while Section 5 presents and discusses the results achieved with the use of our solution. Section 6 presents the conclusions.

2. Related Works

The CORDIC concept was introduced by Volder in 1959 [12], then the solution was extended to calculate elementary functions, including the square root [16]. CORDIC owes its low hardware resource-related requirements to the fact that one iteration may be performed using basic shift and addition commands. A low-complex design methodology is introduced in [4] for the computation of square root ($\sqrt{x}$) and division ($\frac{x}{z}$) using circular CORDIC reuse. This method reduces the area overhead for biomedical applications. Here, the square root is computed using the derivative Newton-Raphson (NR) formula, and a technique consisting in dividing input x into different segments is applied. Solutions [4], [23] also perform micro-rotations to predict rotation directions.

Paper [23] proposed a pipeline-parallel unified CORDIC architecture to perform square root computing and several basic functions using floating point numbers. Article [24] proposes a complex square root computation method independent of angle in the CORDIC style.

Article [11] presents the advantages of the square root CORDIC radix-10 FPGA implementation method. The solution covers both fixed- and floating-point versions with different reconfigurable number of digits, thus different precision versions – according to IEEE 754-2008 standard – are implemented. Therefore, the number of iterations is configurable (13–25). The authors of [25] redesigned the core that employs an iterative non-restoring algorithm that converges closer to the result after every iteration. Articles [9], [11], [18], [26] described the advantages and disadvantages of the various radix-2, radix-4 and radix-10 CORDIC algorithms. These solutions reduce the number of iterations and thus can speed up the calculation process [16], [18].

Article [21] proposes the radix-4 CORDIC algorithm for computing roots and powers of various orders. This solution is divided into three phases, where each stage is completed by a different class of the modified radix-4 CORDIC algorithm. The degree of complexity is reduced by precomputing the scale factor for initial iterations and by employing scaling-free rotations for later iterations. Other papers [27], [28] describe precomputation of rotation direction to achieve a latency improvement, using the radix-4 architecture.

The CORDIC IP core v6.0 developed by Xilinx [29] implements a generalized CORDIC algorithm to iteratively solve trigonometric equations, hyperbolic and square root equations, etc. There are two architectural configurations available: a fully parallel configuration with single-cycle data throughput at the higher expense of silicon die size, and a word serial implementation with multiple-cycle throughput, but with the advantage of low silicon usage.

Alternatively, paper [10] incorporates the square root function into the existing FP multiplication/division fused unit to reduce the hardware resources required. The Taylor series expansion algorithm with powering units, which exhibits the highest performance, was implemented in the hardware.

A parallel CORDIC core with N bit output width has a latency of N cycles and produces a new output every cycle. A word serial CORDIC core with N bit output width has a latency of N cycles and produces a new output every N cycles. In practice, the CORDIC square root function of the Xilinx IP core generator can be useful for the estimation of the DC motor rotor flux [10], [28]. For FPGA implementation, tools provided by FPGA providers are presented in [13]. For example, the Xilinx System Generator (XSG) simulation tool was applied that can easily make the direct translation into hardware of control algorithms without knowing any hardware description language (HDL) for implementing solutions developed in [13]. This is a high-level tool for designing high-performance DSP systems in the Simulink environment.

Similarly, our methodology uses the Xilinx Vitis HLS tool to optimally translate the proposed algorithm into HDL C language. In such a method, the hardware implementation is optimal for any FPGA chips and knowledge of the details of the FPGA architecture is not required.

3. Description of the Proposed Method

In this Section, the application of the CORDIC algorithm for calculating the floating-point square root is described and the open-source code is provided. All known CORDIC square root implementations have been used for integer or fixed point [8]–[10], [14], [26] calculations. The proposed algorithm effectively combines such techniques as convert $\mathrm{fp}\,x$ into an integer i, splitting integer i into a mantissa and an exponent, using the fast bitwise masking operation & (instead of the slow frexpf), combining the converted mantissas and exponents of the result into one number using the same fast bitwise operation (instead of slow ldexpf). It should also be noted that the high accuracy and speed of our algorithm are obtained without the use of the pace-hindering correct rounding operation. Other advantages include the lack of division operations and the presence of only one multiplication opera-

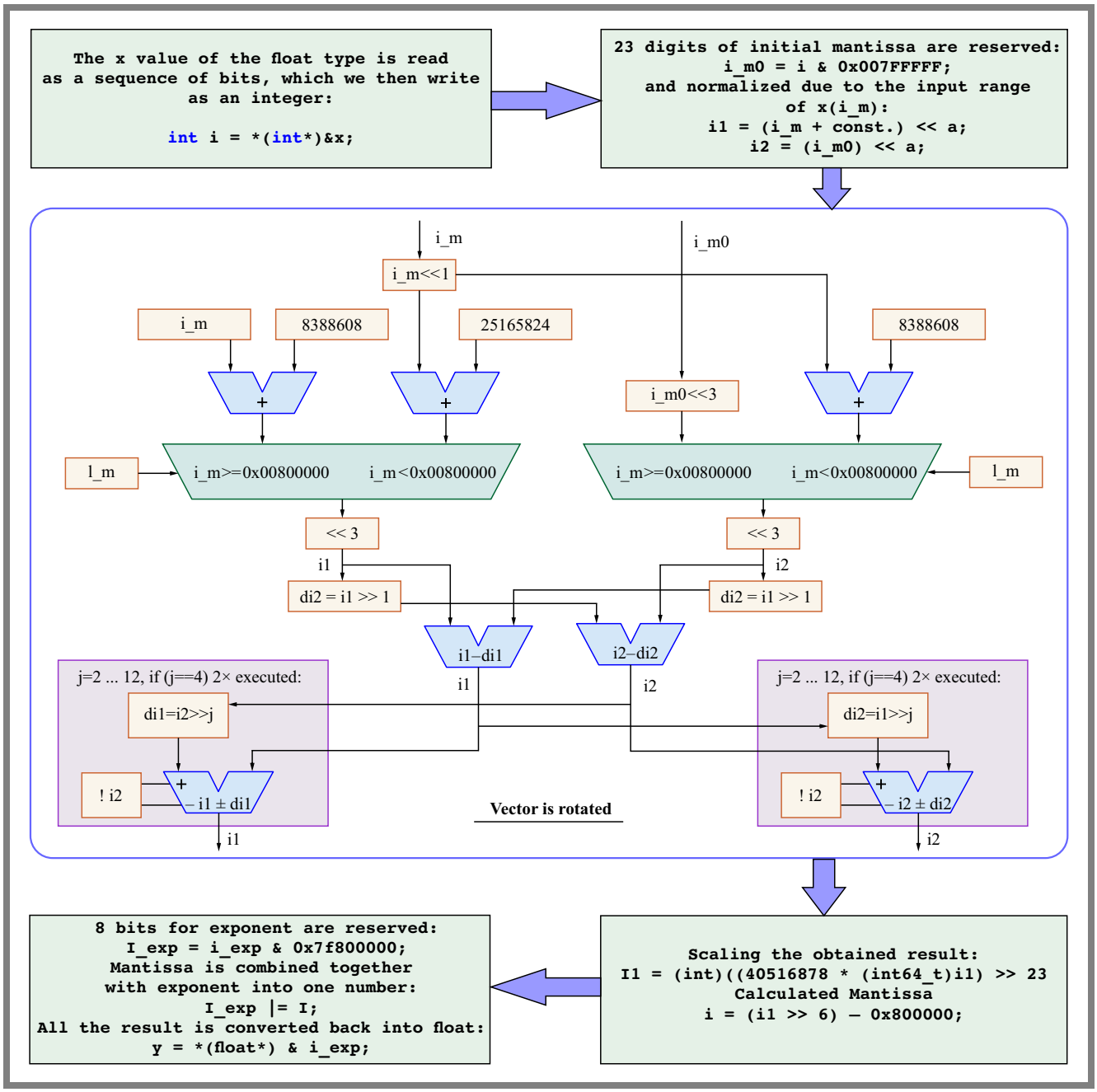

Fig. 1. Low-latency square root CORDIC algorithm.

tion to scale the result. Moreover, the algorithm achieves the expected accuracy in the entire range of normalized floating-point numbers. In this article, we use the following approach to calculate the square root of floating-point numbers [30]. IEEE-754 floating point formats are as follows:

$$-1^{S_x} \cdot M_x \cdot 2^{e_x - bias}. \tag{1}$$

where, for the square root function, $S_x = 0$.
Therefore, it is worth noting that we will use exclusively a biased exponent e_x, and mantissa $M_x \in [1, 2)$. For computing a square root, we have to represent argument x by two separate numbers – exponent and mantissa. Next, the operation presented in Eq. (2) or Eq. (3) is performed:

$$\sqrt{M_x}\, 2^{\frac{e_x - bias}{2}} \quad , \text{ if } e_x - bias \text{ is an even number} \tag{2}$$

and:

$$\begin{aligned} &\sqrt{2M_x}\, 2^{\frac{e_x - bias - 1}{2}} \\ =&\sqrt{2}\, \sqrt{M_x}\, 2^{\frac{e_x - bias - 1}{2}} \quad , \text{ if } e_x - bias \text{ is an odd number .} \end{aligned} \tag{3}$$

From these formulas, one may conclude that the mantissa must be placed in two intervals $M_x \in [1, 2)$ and $2M_x \in [2, 4)$. It can be seen from the above equations that it is necessary to calculate the square root of mantissa M_x. For this reason, we suggest using the CORDIC algorithm. The classical CORDIC algorithm for calculating the square root gives the following equations (vector mode):

$$x_{i+1} = x_i - \sigma_i\, y_i\, 2^{-i} \tag{4}$$

$$y_{i+1} = y_i - \sigma_i\, x_i\, 2^{-i}$$

$$x_0 = M_x + 0.25,\ y_0 = M_x - 0.25$$

$$M_x \in [0.03, 2.33]$$

$$\sqrt{M_x} \approx x_{m+1} P'$$

where P' is the scaling factor, taking into account the repetitions of some iterations (4.4 and 13.13 for example). We have observed that from Eq. (5). It is necessary to expand the range of values to $M_x \in [1, 4)$ instead of 0.03–2.33. Therefore, we proposed to change the constant from 0.25 to 1.0. In this case, we have:

$$x_0 = M_x + 1,\ \ y_0 = M_x - 1\,, \tag{5}$$

$$\sqrt{4\,M_x} \approx 2\,x_{m+1}\,P'\,.$$

In the proposed algorithm, all conversions must be performed in the integer format, as required by CORDIC. The main steps of the new algorithm are presented in Fig. 1 and are summarized as follows:

1) Input argument x is given as a single precision floating point number.
2) x is converted into integer i.
3) Using `0x7f800000` and `0x00ffffff` masks, i is split into two parts: biased exponent i_exp and i_m – the fractional part of the mantissa with the youngest bit of the biased exponent acting as the oldest bit of the mantissa. This bit indicates which range of the mantissa is considered: M_x or $2M_x$. If $i_m \geqslant$ `0x00800000`, then the mantissa is in the $[1, 2)$ range, otherwise in the $[2, 4)$ range.
4) This allows us to set the appropriate values of the exponent of the result – see Eqs. (2) and (3).
5) The initial values of numbers i_1 and i_2 are introduced, corresponding to variables x_0 and y_0 of the CORDIC algorithm for SQRT calculation; see Eq. (5). Depending on the range of the mantissa, M_x or $2M_x$, these are the initial values of x_0 and y_0. These constants are the newly proposed elements of the presented algorithm.
6) In addition, in the CORDIC algorithm, three additional bits are used to increase the accuracy of calculating the square root of the mantissa, up to the available 23-24 bits of mantissa available in float-type numbers.
7) The result obtained is scaled by integer multiplication: $(\mathrm{int})((40516878 * (int64_t) i_1) >> 23)$.

4. Proposed Algorithm

The aforementioned methodology is implemented in the algorithm presented in Fig. 2.

```
float Sqrt_Cordic_1 (float x)
{ float y;
 int32_t i_m, i_exp;
 int32_t i, i1, i2, i_m0, i_m1, di1, di2,j;
 i = *(int*)&x;
 i_exp = i & 0x7f800000;
 i_m = i & 0x00fffff;
 i_m0 = i & 0x007fffff;
  if (i_m>= 0x00800000){
      i1=(i_m+8388608)<<3; i2=i_m0<<3}
     else{i1=((i_m<<1)+25165824)<<3;
       i2=((i_m<<1)+ 8388608)<<3;}
  di1 = (i2 >> 1); di2 = (i1 >> 1);
  i1 = i1 - di1; i2 = i2 - di2;
 for (j = 2; j <=12; j++ )
  {di1 = (i2 >> j); di2 = (i1 >> j);
  if (i2 >= 0) { i1 = i1 - di1; i2 = i2 - di2;}
     else { i1=i1+di1; i2=i2+di2; }
  if (j= =4){ di1 = (i2 >> j); di2 = (i1 >> j);
    if (i2 >= 0) { i1 = i1 - di1; i2 = i2 - di2;}
      else { i1=i1+di1; i2=i2+di2; }
  }
  }
i1 = (int)(( 40516878* (int64_t)i1)>> 23) ;
i=(i1>>6)- 8388608;
i_exp=(i_exp + 0x3f800000)>>1;
i_exp = i_exp &0x7f800000;
i_exp |= i;
y = *(float*)&i_exp;
  return y;
}
```

Fig. 2. Square root floating-point function algorithm.

The proposed floating point CORDIC function was implemented on several families of FPGAs. For this purpose, the Xilinx Vitis HLS automatic synthesis and implementation software were used, generating Verilog code at the RTL level. The project was implemented by utilizing basic FPGA resources, such as LUT-based logic, multipliers from DSP blocks, flip-flops (FFs), etc.

The algorithm generates results from 2 clock cycles for the fastest FPGAs (i.e. Versal, Virtex-7 Ultra Plus, Kintex-7 Ultra Plus), up to 8 clocks for the slowest FPGA Spartan-7. The CORDIC function written in C, as presented in this paper, is easily implementable on FPGA chips and requires a small amount of resources.

The Xilinx FPGA 7 family is optimized for low-power applications requiring serial transceivers, fast DSP and high logic throughput. The logic is based on real 6-input LUT technology, configurable as distributed memory. DSP slices are designed with a 25×18 multiplier, a 48-bit accumulator, and a pre-adder for high performance filtering, including optimized symmetric coefficient filtering [31].

The Artix-7 family includes up to 215 K logic cells, 13 Mb RAM block, and 740 DSP slices. The more sophisticated families, i.e. Kintex-7 and Virtex 7, include: in the case of Kintex-7 – up to 478 K logic cells, 34 Mb RAM block, and 1920 DSP slices. In the case of Virtex-7 – up to 1955 K logic cells, 68 Mb RAM block, and 3600 DSP slices.

The AMD Xilinx Zynq UltraScale+ MPSoC family is based on the UltraScale MPSoC architecture. This series integrates a 64-bit quad-core or dual-core Arm Cortex-A53 and dual-core Arm Cortex-R5F-based processing system (PS) and Xilinx programmable logic (PL) UltraScale architecture in

Tab. 1. Results of logic synthesis of SQRT FP CORDIC using Xilinx Vitis HLS tool.

FPGA	T [ns]	No. of cycles	No. of DSP	No. of LUT	LUT usage	No. of FF	FFs usage
Artix-7	70	7	3	2878	6.7%	642	35.96%
Kintex-7	50	5	3	2870	7.00%	493	0.60%
Kintex-7U	40	4	2	2871	0.43%	395	0.03%
Kintex-7 UP	20	2	2	2860	1.76%	252	0.08%
Spartan-7	80	8	3	2886	36.06%	697	4.63%
Virtex-7	50	5	3	2870	1.40%	438	0.11%
Virtex UP	20	2	2	2860	0.72%	155	0.02%
Zynq-7000	70	7	3	2878	16.35%	607	1.72%
ZynqUP	30	3	2	2866	3.26%	269	0.15%
Versal	20	2	2	2698	2.40%	230	0.01%

Tab. 2. Results of FPGA implementation of SQRT FP CORDIC by Xilinx Vitis HLS.

FPGA	T [ns]	No. of cycles	No. of DSP	DSP usage	No. of LUT	LUT usage	No. of FF	FF usage
Artix-7	70	7	4	8.88%	982	12.28%	416	2.60%
Kintex-7	50	5	4	1.70%	862	2.10%	262	0.32%
Kintex-7U	40	4	2	0.04%	862	0.13%	210	0.02%
Kintex-7 UP	20	2	2	0.15%	1042	0.64%	199	0.06%
Spartan-7	80	8	4	20.00%	976	12.2%	464	2.9%
Virtex-7	50	5	3	0.27%	861	0.42%	232	0.05%
Virtex UP	20	2	2	0.09%	858	0.22%	82	0.01%
Zynq-7000	70	7	4	5.0%	1099	6.24%	468	1.33%
ZynqUP	30	3	2	0.27%	905	1.03%	136	0.08%
Versal	20	2	2	0.1%	810	0.72%	122	0.007%

a single device. Also included are on-chip memory, multiport external memory interfaces, and a large set of peripheral connectivity interfaces. The Zynq UltraScale+ MPSoC family includes up to 1.14 M logic cells, in particular 522 K CLB LUT, 1 M CLB flip-flops, 984 Mb RAM block, and 2 M DSP slices [32].

5. Achieved Results

The maximum negative and positive errors achieved by the proposed method are –1.7001956E–07 and 1.0053241E–07, respectively. Additionally, we can observe four special cases for normalized numbers:

- zero $x = 0.0$; $y = 7.6664669522108749\text{E–}20$,
- min $x = 1.1754942106924410754870029\text{E–}38$; $y = 1.0842021078620191\text{E–}19$,
- max $x = 3.4028234663852885981170420\text{E+}38$, $y = 1.8446742974197924\text{E+}19$,
- $x = \infty$; $y = 1.8446744073709552\text{E+}19$.

We checked the relative errors over the full range of normalized single-precision floating-point numbers using the $\mathrm{nextafterf}()$ function for the C++ code of our algorithm, comparing the results with the `sqrt` function of the `cmath` library. $\mathrm{nextafterf(a, b)}$ is a function defined in the C++ cmath library. Return the next representable value after x in the y direction. The relative error was calculated as:

$$dr = (\text{double})\frac{y}{\sqrt{x}} - 1\ .$$

The y results from the C++ code were compared with the results obtained in FPGA – the integer number results were always the same.

Table 1 presents the execution time of the proposed square root floating point (FP) CORDIC algorithm and FPGA resources utilized after logical synthesis on the chips. Table 2 collects the same results after FPGA implementation by means of the Xilinx Vitis HLS tool that provides optimization of the resources used.

First, one of the widely used FPGA Artix-7 chips was chosen. The clock frequency was set to a default value of 100 MHz,

because many other FPGAs can operate at this frequency. This makes it easier to compare the achievements of FPGAs originating from different families. The execution time and the resources used after the logical synthesis using the Xilinx Vitis tool for Artix-7 are presented in Tab. 1. Our floating point CORDIC function required 7 clock cycles to generate output, i.e. 70 ns.

Implementing the proposed function on the smallest available chip of the Artix-7 family (`xc7a12t-cpg238-3`) required 4 (8.8%) built-in DSP blocks, 982 (12.28%) LUTs, and 416 (2.6%) flip-flops. Xilinx does not disclose the results of the own implementation of the CORDIC v. 6.0 IP on Artix-7 FPGAs [29].

Next, the Kintex-7 FPGA was tested. The proposed function took 5 clock cycles (50 ns) to complete. Implementation of the algorithm on the smallest chip of the Kintex-7 family (`xc7k70t-fbv676-3`) required 4 (1.7%) built-in DSP blocks, 862 (2.10%) LUTs, and 262 (0.32%) flip-flops. According to Xilinx [29], their square root in CORDIC IP occupied 2184 LUTs, and 1328 flip-flops on the Kintex-7 chip, which is approximately 2.5 times more LUTs, and 5 times more flip-flops than in the proposed solution.

Furthermore, we implemented the algorithm on the Kintex-7 Ultra (Kintex-7U) FPGA. The execution took 4 clock cycles (40 ns) on the smallest chip of the Kintex-7 Ultra family (`xcku115-flva1517-3-e`) and required 2 (0.04%) built-in DSP blocks, 862 (0.13%) LUTs, and 210 (0.02%) FFs. According to Xilinx [29], their SQRT function occupied 2278 LUTs and 1336 flip-flops, which is approximately three times more than for the proposed solution.

The last FPGA variant of the Kintex-7 family was Kintex-7 Ultra Plus (Kintex-7 UP). In this case, the algorithm was executed in 2 clock cycles (20 ns) only. It means that implementation on the smallest chip of the Kintex-7 Ultra Plus family (`xcku3p-sfvb784-3-e`) required 2 (0.15%) built-in DSP blocks, 1042 (0.64%) LUTs, and 199 (0.06%) FFs. The Xilinx solution uses 2208 LUTs and 1334 FFs. This is also approximately 2.12 times LUTs more and 6.7 times more flip-flops than for the proposed solution.

Tables 1 and 2 illustrate that the proposed function implemented on different FPGAs of the Kintex-7 family occupied a similar amount of resources.

Moreover, we also tested the Spartan-7 FPGA (`xc7s15ftgb196-2`), a chip with a slightly different and older architecture when compared to the previous family. In this case, the proposed function achieved the worst result of 8 clock cycles (80 ns), using a small portion of the resources available, i.e. 4 (20%) built-in DSP blocks, 976 (12.2%) LUTs, and 464 (2.9%) flip-flops. This result is a consequence of the lower quantity of resources in this chip: 20 DSP, 8000 LUTs, and 16000 FFs. Xilinx does not disclose the results achieved while implementing CORDIC v. 6.0 IP on Spartan-7 FPGAs [29].

The Virtex-7 series FPGA is optimized for the best performance and capacity and is used in the verification `xc7vx330t-ffv1761-3` chip, which offers abundant resources and achieves top performance in the Xilinx FPGA family. The execution time of the proposed function was 5 clock cycles (50 ns) with the usage of only 3 (0.27%) built-in DSP blocks, 861 (0.42%) LUTs, and 232 (0.05%) FFs. According to Xilinx [29], their SQRT solution occupied 2177 LUTs and 1328 FFs, which is approximately 2.5 times more LUTs, and 5.7 times more FFs than in the case of our solution.

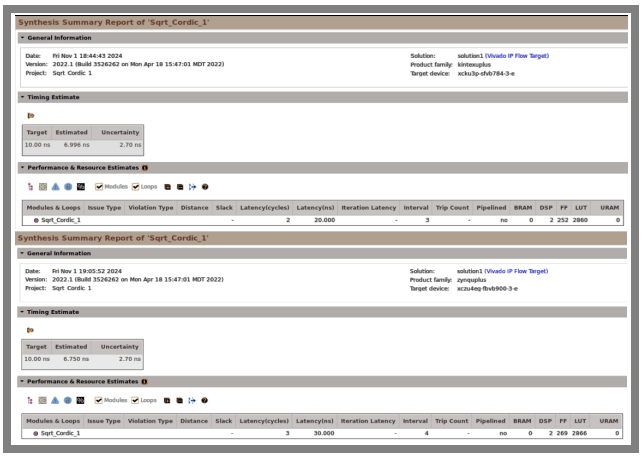

Fig. 3. Results of the synthesis of Xilinx Vitis HLS on: Kintex Ultra Plus FPGA (top) and Zynq Ultra Plus FPGA (bottom).

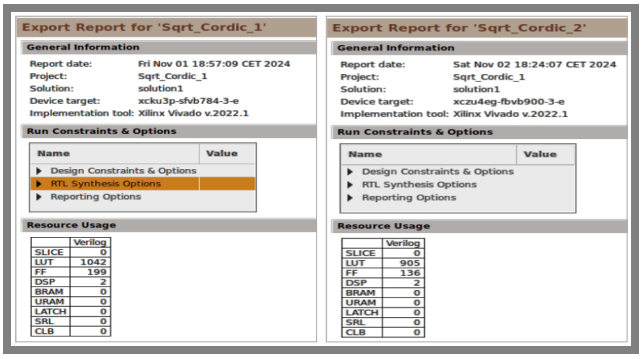

Fig. 4. Results of implementation of Xilinx Vitis HLS on Kintex Ultra Plus FPGA (left) and on Zynq Ultra Plus FPGA (right).

The Virtex-7 Ultra Plus (`xcvu3pCIV-ffvc1517-3-e`) algorithm took only two clock cycles (20 ns) to complete. The function occupies merely 2 DSP blocks (0.09%), a low number of 858 LUTs (0.22%), and 82 flip-flops (0.01%). Compared to Xilinx [29], their SQRT IP occupied 2242 LUTs and 1342 flip flops, which is approximately 2.6 times more LUTs and 16 times more FFs than in the proposed solution.

On Zynq Ultra Plus (`xczu4eg-fbvb900-3-e`) FPGA the algorithm required only 2 built-in DSP blocks (0.27%), 905 LUTs (1.03%), and 136 FFs (0.08%). The function was executed within 3 clock cycles (30 ns), while Xilinx [29] SQRT occupied 2177 LUTs, and 1330 FFs, which is about 2.4 times more LUTs, and 9.7 times more flip-flops compared to proposal.

We also tested an older version of the Zynq-7000 (`xc7z010-clg225-3`) FPGA chip, with the execution taking 7 clock cycles (70 ns) and using 4 built-in DSP blocks (5%), 1099 LUTs (6.24%) and 468 flip-flops (1.33%). Compared to Xilinx [29], their SQRT occupied 2177 LUTs and

1330 flip-flops, which is approximately 2 times more LUTs, and 2.8 times more FFs than in our proposal.

Finally, we implemented the algorithm in the most advanced chip family, namely the Versal.

The implementation required a negligible amount of resources of the smallest Versal chip (`xcvc1902-viva1596-3HP-e-S`), i.e. only 2 DSP blocks (0.1%), 810 (only 0.72%) LUTs, and 122 FFs (0.007% of all FFs available). The algorithm required only 2 clock cycles (20 ns) to complete. In comparison to Xilinx, their solution occupied 1968 LUTs and 1454 FFs in this FPGA, which is approximately 2.43 times more LUTs and 11.9 times more flip-flops than in the case of our proposal. All the results of logic synthesis performed using the Xilinx Vitis HLS software are presented in Tab. 1, while Tab. 2 illustrates the results of the FPGA implementations of the proposed CORDIC function.

These results have been optimized due to the architectural details of each selected FPGA. Details concerning the time of the signal's propagation through a CLB for the basic families of FPGAs, retrieved from Xilinx data sheets, are presented in Tab. 3. Data on the operating speed of FPGAs utilized in the experiments highlights the time efficiency of the proposed algorithm in the context of its complexity.

Example results of Xilinx Vitis HLS synthesis on Kintex Ultra Plus FPGA are shown in Fig. 3, while results of Xilinx Vitis HLS implementation on Kintex Ultra Plus FPGA are presented in Fig. 4.

The usage of FPGA resources required to implement CORDIC-based modules usually exceeds 1100 LUTs and 1000 flip-flops. Our solution achieved a result that was 20% better. The CORDIC structure [33] was of the combined iterative three-stage or multistage variety and required slightly more resources than our solutions inside the Kintex-7 FPGA. However, only the sum of LUTs and flip-flops together is given, and unlike our algorithm, those need ROM memory. The execution time ranges from 60 ns to 190 ns, meaning it is longer than the time achieved by us (50 ns on Kintex-7).

An FPGA implementation of the CORDIC floating point SQRT performed on Virtex-7 in the iterative pipeline-parallel version, as presented in [23], occupied 708 LUTs. However, it required weighting the scale factors by applying an additional 25-bit fixed-time expensive multiplication to generate final results. The implementation of the complex SQRT method proposed in [24] on the Virtex-6 FPGA occupied 6852 LUTs and generated a latency of 38 clock cycles. The square root calculated on the Virtex-7 Ultra Plus FPGA by the Radix-10 CORDIC algorithm presented in [11] required from 1193 (7 digit version) to 5796 LUTs (34 digits version) and from 339 (7 digits) up to 1481 (34 digits) flip-flops, depending on the number of precision digits. Latency ranged from 10 clock cycles for 7 digits to 37 clock cycles. It was the worst result when compared to the application of our proposal on the same FPGA family (2 clock cycles).

Our solution also consumed approximately 25% less FPGA resources. As a further example, a faster radix-4 root CORDIC algorithm with a 40-bit precision level required 9324 LUTs when implemented on the Virtex-6 FPGA [21]. This is a hyperbolic CORDIC utilizing Taylor's approximation. The concurrent radix-4 CORDIC solution proposed in [26] was implemented in the Spartan-6 FPGA and occupied 6840 LUTs on this FPFA, with a latency of 68 ns. For the older FPGA version (Spartan-3E), 4508 LUTs were used and latency equaled 80 ns.

Despite the lower amount of hardware resources required, the relatively high latency of [10] was achieved for his high-performance SQRT circuit based on the Taylor series. The authors of [25] focused on maximizing operating frequency as well as reducing static and dynamic power levels. However, their implementation of the FP SQRT occupied, for instance, 804 basic and 971 enhanced LUTs of the Virtex-5 FPGA. Our result of 861 LUTs achieved on the Virtex-7 FPGA ranks us in the middle of their range.

One may notice that sophisticated solutions require similar to proposed bit of precision amount of FPGA resources and a higher number of clock cycles to complete specific functions.

Implementation of fixed-point SQRT CORDIC solutions is usually more frequent in FPGAs. Therefore, it is easier to find many more publications about fixed-point algorithms implemented in FPGAs. However, the proposed FP solution often achieves a lower latency level than many of its fixed-point counterparts, displaying a similar or lower demand for FPGA resources.

Tab. 3. Signal propagation time through configurable logic block.

FPGA	Artix-7	Kintex-7	Spartan-7	Virtex-7	Zynq-7000
Combinatorial [ns]	0.94	0.58	1.05	0.58	0.94
Sequential [ns]	0.47	0.32	0.53	0.32	0.47
CLB set and hold [ns]	0.59	0.36	0.66	0.36	0.59
Set/reset [ns]	0.53	0.52	0.78	0.52	0.53
DSP I/O [ns]	4.06	3.44	4.65	3.44	4.06

6. Conclusions

In this article, a new algorithm is presented for CORDIC square root computation for FP numbers. The original methodology of converting floating numbers to integers and back allows to optimize the usage of FPGA hardware resources and lowers the efficiency of the calculation. We achieved a very short computation time of two clock cycles on Ultra Scale Plus Xilinx FPGAs from the Kintex-7 and Virtex-7 series. The same result was achieved on the Versal FPGA. The usage of FPGA resources by the proposed solu-

tion is similar to or lower than that of the more sophisticated optimization methods presented in the literature.

The authors' contribution to the field can be summarized as follows:

- FPGA floating point CORDIC SQRT circuit for normalized numbers in the single precision IEEE754 format,
- Maximum relative error of 1.7E–7,
- Relatively simple theory, easily implementable on FPGAs,
- Low average implementation latency on widespread FPGAs,
- Very low implementation latency on Ultra Plus Families of FPGAs,
- No division operation, only one integer multiplication operation (to scale the result),
- Decent accuracy over the entire range of normalized FP numbers,
- Lower or similar utilization of FPGA resources compared with other solutions.

Our future research will focus on methods relied upon to perform fixed-point CORDIC computations of several basic functions. We currently work on a new approach to angle recoding allowing to flexibly adjust the memory table size and the number of CORDIC iterations.

References

[1] L. Moroz, V. Samotyy, M. Wegrzyn, and U. Dzelendzyak, "Efficient Floating-point Square Root and Reciprocal Square Root Algorithms", *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Cracow, Poland, 2021 (https://doi.org/10.1109/IDAACS53288.2021.9660872).

[2] A. Hasnat *et al.*, "A Fast FPGA Based Architecture for Computation of Square Root and Inverse Square Root", *Devices for Integrated Circuit (DevIC)*, Kalyani, India, 2017 (https://doi.org/10.1109/DEVIC.2017.8073975).

[3] Z. Kokosinski *et al.*, "Fast and Accurate Approximation Algorithms Computing Floating Point Square Root", *Numerical Algorithms*, 2024 (https://doi.org/10.1007/s11075-024-01932-7).

[4] S. Mopuri, S. Bhardwaj, and A. Acharyya, "Coordinate Rotation-based Design Methodology for Square Root and Division Computation", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, pp. 1227–1231, 2019 (https://doi.org/10.1109/TCSII.2018.2878599).

[5] R. Shukla and K.C. Ray, "Low Latency Hybrid CORDIC Algorithm", *IEEE Transactions on Computers*, vol. 63, pp. 3066–3078, 2014 (https://doi.org/10.1109/TC.2013.173).

[6] M.D. Ercegovac and T. Lang, *Division and Square Root Digit-recurrence Algorithms and Implementations*, Norwell: Kluwer Publishers, 240 p., 1994 (ISBN 9780792394389).

[7] Y.H. Hu and S. Naganathan, "An Angle Recoding Method for CORDIC Algorithm Implementation", *IEEE Transactions on Computers*, vol. 42, pp. 74–79, 1993 (https://doi.org/10.1109/12.192217).

[8] E. Antelo, T. Lang, and J. Bruguera, "Very-high Radix Circular CORDIC: Vectoring and Rotation/vectoring", *IEEE Transactions on Computers*, vol. 49, pp. 727–739, 2000 (https://doi.org/10.1109/12.863043).

[9] E. Antelo, J. Villalba, J.D. Bruguera, and E. Zapata, "High Performance Rotation Architectures Based on Radix-4 CORDIC Algorithm", *IEEE Transactions on Computers*, vol. 46, pp. 855–870, 1997 (https://doi.org/10.1109/12.609275).

[10] T.-J. Kwon and J. Draper, "Floating-Point Division and Square Root Implementation using a Taylor-series Expansion Algorithm with Reduced Look-up Tables", *2008 51st Midwest Symposium on Circuits and System*, Knoxville, USA, 2008 (https://doi.org/10.1109/MWSCAS.2008.4616959).

[11] Martín Vázquez, Marcelo Tosini, Lucas Leiva., "Radix-10 Restoring Square Root for 6-input LUTs Programmable Devices", *Circuits Systems and Signal Processing*, vol. 40, pp. 2335–2360, 2021 (https://doi.org/10.1007/s00034-020-01571-y).

[12] J.E. Volder, "The CORDIC Trigonometric Computing Technique", *IEEE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, 1959 (https://doi.org/10.1109/TEC.1959.5222693).

[13] J.-G. Mailloux, S. Simard, and R. Beguenane, "FPGA Implementation of Induction Motor Vector Control using Xilinx System Generator", *6th WSEAS International Conference on Circuits, Systems, Electronics, Control & Signal Processing*, Cairo, Egypt, 2007.

[14] M. Garrido, P. Källström, M. Kumm, and O. Gustafsson, "CORDIC II: A New Improved CORDIC Algorithm", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, pp. 186–190, 2016 (https://doi.org/10.1109/TCSII.2015.2483422).

[15] S. Srinivasan *et al.*, "Split-path Fused Floating Point Multiply Accumulate (FPMAC)", *21th IEEE Symposium on Computer Arithmetic*, Austin, USA 2013 (https://doi.org/10.1109/ARITH.2013.32).

[16] J.S. Walther, "A Unified Algorithm for Elementary Functions", *Proc. of AFIPS Joint Computer Conferences*, vol. 38, pp. 385–389, 1971 (https://doi.org/10.1145/1478786.1478840).

[17] S. Wang, V. Piuri, and E.E. Swartzlander, "Hybrid CORDIC Algorithms", *IEEE Transactions on Computers*, vol. 46, no. 11, pp. 1202–1207, 1997 (https://doi.org/10.1109/12.644295).

[18] P.-T. Vo-Thi, T.-T. Hoang, C.-K. Pham, and D.-H. Le, "A Floating-point FFT Twiddle Factor Implementation Based on Adaptive Angle Recoding CORDIC", *2017 International Conference on Recent Advances in Signal Processing Telecommunications & Computing (SigTelCom)*, Da Nang, Vietnam, 2017 (https://doi.org/10.1109/SIGTELCOM.2017.7849789).

[19] A. Madisetti, A.Y. Kwentus, and A.N. Willson, "A 100 MHz, 16-b, Direct Digital Frequency Synthesizer with 100-dBc Spurious-free Dynamic Range", *IEEE Journal of Solid-State Circuits*, vol. 34, no. 8, pp. 1034–1043, 1999 (https://doi.org/10.1109/4.777100).

[20] D. Timmermann, H. Hahn, and B. Hosticka, "Low Latency Time CORDIC Algorithms", *IEEE Transactions on Computers*, vol. 41, pp. 1010–1015, 1992 (https://doi.org/10.1109/12.156543).

[21] M. Woźniak *et al.*, "Radix 4 CORDIC Algorithm Based Low Latency and Hardware Efficient VLSI Architecture for *N*th Root and *N*th Power Computations", *Scientific Reports*, vol. 13, art. no. 20918, 2023 (https://doi.org/10.1038/s41598-023-47890-3).

[22] R. Dutt and A. Acharyya, "Low-complexity Square-root Unscented Kalman Filter", *Circuits, Systems, and Signal Processing*, vol. 42, pp. 6900–6928, 2023 (https://doi.org/10.1007/s00034-023-02437-9).

[23] B. Li *et al.*, "A Unified Reconfigurable Architecture Based on CORDIC Algorithm Floating-point Arithmetic", *2017 International Conference on Field Programmable Technology (ICFPT)*, Melbourne, Australia, 2017 (https://doi.org/10.1109/FPT.2017.8280166).

[24] S. Mopuri and A. Acharyya, "Low-complexity and High-speed Architecture Design Methodology for Complex Square Root", *Circuits, Systems, and Signal Processing*, vol. 40, pp. 5759–5772, 2021 (https://doi.org/10.1007/s00034-021-01738-1).

[25] S. Suresh, S.F. Beldianu, and S.G. Ziavras, "FPGA and ASIC Square Root Designs for High Performance and Power Efficiency", *2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors*, Washington, USA, 2013 (https://doi.org/10.1109/ASAP.2013.6567588).

[26] M.A. Darshan, "A High Performance and Low Latency FPGA Implementation of CORDIC Algorithm", *International Journal of Scientific & Engineering Research*, vol. 4, no. 8, 2013 (ISSN 22295518).

[27] T.-B. Juang, S.-F. Hsiao, and M.-Y. Tsai, "Para-CORDIC: Parallel CORDIC Rotation Algorithm", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, pp. 1515–1524, 2004 (https://doi.org/10.1109/TCSI.2004.832734).

[28] T. Juang, "Low Latency Angle Recoding Methods for the Higher Bit-width Parallel CORDIC Rotator Implementations", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, pp. 1139–1143, 2008 (https://doi.org/10.1109/TCSII.2008.2002566).

[29] Xilinx, "CORDIC v6.0 LogiCORE IP Product Guide", 2021.

[30] F. de Dinechin, M. Joldes, B. Pasca, and G. Revy, "Multiplicative Square Root Algorithms for FPGAs", *2010 International Conference on Field Programmable Logic and Applications*, Milan, Italy, 2010 (https://doi.org/10.1109/FPL.2010.112).

[31] AMD Xilinx, "7 Series FPGAs Data Sheet: Overview DS180 (v2.6.1)", product specification, 2020.

[32] AMD Xilinx, "Zynq UltraScale+ MPSoC Data Sheet: Overview DS891 (v1.10)", product specification, 2022.

[33] M. Qin *et al.*, "A Low-latency RDP-CORDIC Algorithm for Real-time Signal Processing of Edge Computing Devices in Smart Grid Cyber-physical Systems", *Sensors*, vol. 22, art. no. 7489, 2022 (https://doi.org/10.3390/s22197489).

Mariusz Węgrzyn, Ph.D.
Faculty of Electrical and Computer Engineering
https://orcid.org/0000-0002-6938-2954
E-mail: mariusz.wegrzyn@pk.edu.pl
Cracow University of Technology, Cracow, Poland
https://www.pk.edu.pl

Stepan Voytusik, D.Sc.
Department of Information Technology Security
https://orcid.org/0000-0003-4234-3303
E-mail: voytusik.b@gmail.com
Lviv Polytechnic National University, Lviv, Ukraine
https://lpnu.ua/en

Nataliia Gavkalova, Prof.
Faculty of Mechanical and Industrial Engineering
https://orcid.org/0000-0003-1208-9607
E-mail: nataliia.gavkalova@pw.edu.pl
Warsaw University of Technology, Warsaw, Poland
https://eng.pw.edu.pl

Task Offloading and Scheduling Based on Mobile Edge Computing and Software-defined Networking

Fatimah Azeez Rawdhan

Mustansiriyah University, Baghdad, Iraq

https://doi.org/10.26636/jtit.2025.1.1941

Abstract — When integrated with mobile edge computing (MEC), software-defined networking (SDN) allows for efficient network management and resource allocation in modern computing environments. The primary challenge addressed in this paper is the optimization of task offloading and scheduling in SDN-MEC environments. The goal is to minimize the total cost of the system, which is a function of task completion lead time and energy consumption, while adhering to task deadline constraints. This multi-objective optimization problem requires balancing the trade-offs between local execution on mobile devices and offloading tasks to edge servers, considering factors such as computation requirements, data size, network conditions, and server capacities. This research focuses on evaluating the performance of particle swarm optimization (PSO) and Q-learning algorithms under full and partial offloading scenarios. Simulation-based comparisons of PSO and Q-learning show that for large data quantities, PSO is more cost efficient than the other algorithms, with the cost increase equaling approximately 0.001% per kilobyte, as opposed to 0.002% in the case of Q-learning. As far as energy consumption is concerned, PSO performs 84% and 23% better than Q-learning in the case of full and partial offloading, respectively. The cost of PSO is also less sensitive to network latency conditions than GA. Furthermore, the results demonstrate that Q-learning offers better scalability in terms of execution time as the number of tasks increases, and exceeds the outcomes achieved by PSO for task loads of more than 40. Such observations prove that PSO is better suited for large data transfers and energy-critical applications, whereas Q-learning is better suited for highly scalable environments and large numbers of tasks.

Keywords — energy efficiency, MEC, PSO, Q-learning, scalability, scheduling, SDN

1. Introduction

The rapid development of mobile devices and the increasing importance of computationally-intensive services present numerous difficulties to mobile computing [1]. Mobile edge computing (MEC) has been developed to handle the problems in question by providing the necessary computational capabilities closer to the needs [2]. At the same time, software-defined networking (SDN) significantly altered the nature of networks by offering the ability to manage them through a logically centralized control interface separated from the data plane of the forwarding devices, thus ensuring flexibility and programmability for the management of the networks [3].

The combination of MEC and SDN is promising to be a solution that could improve the efficiency of mobile computing even further. SDN provides centralized control of the network, while MEC systems will be able to make better decisions concerning task offloading and resource management.

MEC provides a function known as task offload which involves reallocation of computational tasks from mobile devices that are constrained in terms of resources to edge servers with a relatively higher quantity of resources available. Efficient offloading of tasks is complicated and requires that a number of factors be considered, such as network availability, server capacity, energy consumption time, and deadlines [4].

Classical approaches to offloading have drawbacks when applied in mobile networks when it comes to achieving efficient resource allocation, especially when the task-related requirements vary. However, due to the existence of a large number of different mobile devices and edge servers, the problem of task offloading also faces another challenge. Mobile equipment can produce different levels of computation and consume various amounts of power, while edges servers might vary in their computational capacity and available resources [5].

This heterogeneity makes task offloading and scheduling a more complex issue, as the technique should be able to accommodate a wide range of device characteristics and network conditions. Due to the employment of new machine learning techniques, there are new prospects for solving these issues. Some reinforcement learning techniques, such as Q-learning algorithms, have been found to be effective in optimizing decisions in dynamic environments [6]. Similarly, other bio-inspired metaheuristic algorithms, such as particle swarm optimization (PSO), have been used to solve complex problems, such as scheduling [7].

The framework proposed in this paper introduces a new solution based on integrating SDN, MEC, and state-of-the-art machine learning approaches for efficient offloading and scheduling of tasks in the context of MEC. The proposed approach takes advantage of the global view of the network that SDN provides to collect information about the conditions in the network and the availability of resources in real-time. This

information is relied upon by a combination of Q-learning and PSO algorithms to determine the resource allocation plan.

The action selection approach the presented solution is based on includes also a Q-learning component, thus improving performance of the system by monitoring changes in network conditions and task characteristics over time. This allows to make dynamic decisions as to when the fully or partially off-loaded approaches should be used, depending on the conditions of the network and task-related demands. Furthermore, the proposed technique uses the PSO algorithm to optimize the schedule of the performing offloaded tasks in multiple edge servers in order to prevent resource waste and uneven load distribution. This framework also includes a dynamic cost model taking into account energy consumption, processing time, network latency, and the completion lead time required for a specific task.

The remainder of this paper is organized as follows. Section 2 presents the relevant literature on multiedge computing, software-defined networking, and various task offloading approaches. Section 3 describes the model of the system and formulates the problem. Section 4 presents a hybrid Q-learning and PSO-based offloading and scheduling plan. In Section 5, a detailed description of the simulation environment and the results that were obtained are presented. Lastly, conclusion are drawn and suggestions for future research are presented in Section 6.

2. Related Works

The authors of [8] proposed an integrated approach to task and resource allocation for MEC in an IoT network, using deep reinforcement learning. Implementation of a deep Q network leads to an overall energy output decrease of 15%, in addition to a 10% increase in output rate, as opposed to conventional methods. The scheme has shown adequate performance for various types and densities of networks and tasks. [9] proposed a metaheuristic approach to task scheduling for MEC based on a combination of genetic and PSO algorithms. The approach demonstrated an 18% reduction in total latency along with a 12% better efficiency of resources for various types of work. One of the algorithm's main strengths was being able to calibrate itself to the varying capabilities of edge servers.

The authors of [10] proposed a federated learning-based method to design the offload of protective tasks in an SDN-supported MEC environment. They obtained 22% less network overhead and 17% better privacy of control compared to the conventional centralized learning technique. The said approach offered great scalability as well. In other words, it was capable of addressing scenarios with a significant number of edge devices. A multi-objective optimization in MEC for energy-sensitive task offloading was proposed in [11], where an improved version of ant colony optimization was employed. The approach worked towards attaining near-optimal solutions for both power consumption and time required to complete a task, with energy consumption reduced by 20% and with a 14% increase in accomplishments per task under dynamic networks.

In article [12], the authors developed a new edge intelligence concept that combines blockchain and deep reinforcement learning to ensure safe and optimal task relocation in MEC. This strategy indicated that there is a 25% improvement in security measures and that the latency of the end-to-end technique is 16% lower than that of traditional techniques. That overarching framework was shown to work well in scenarios ranging from low to high levels of distrust among the edge nodes.

The authors of [13] proposed a context-aware task offloading scheme for MEC in a 5G network based on LSTM and the Q-learning algorithm. Their approach achieved a remarkably high-level of improvement in QoS satisfaction (19%) and a 13% reduction in energy consumption. According to more recent investigations, the scheme proved to be more efficient with regard to forecasting and managing user mobility.

2.1. Limitations of Related Research Work

In the existing literature concerned with task offloading in MEC and SDN, the following limitations are observed. Most of the existing approaches apply classic algorithms that were not capable of adjusting to the dynamic natures of network conditions and task load variations at all times. This led, in many cases, to suboptimal resource allocation and higher latency. Additionally, some algorithms incorporate reinforcement learning methods. However, those may not aim to increase both energy efficiency and performance scalability in most applications, especially when the amounts of resources are limited. The presented approach will integrate both PSO and Q-learning, so that energy efficiency will be combined with scalability of resources as the number of tasks increases. Furthermore, the new method introduces a more flexible task sharing solution that may fit both complete and partial sharing techniques, offering a higher level of flexibility.

3. System Model

In this section, we present the mathematical models and equations used in our MEC-SDN integrated system for task offloading and scheduling. The architecture of an SDN-based MEC environment consists of three main layers (Fig. 1):

- **Cloud computing layer.** It is located at the top and comprises centralized cloud facilities with a core network connected to cloud data centers.
- **Edge computing layer.** The middle layer is located between the cloud and the infrastructure and has an SDN global controller to control the entire network. It is implemented in the form of numerous edge computing zones with their own SDN local controllers. Each zone includes MEC servers used as computation facilities located at the edge of the network and OpenFlow switches for SDN-based network management.

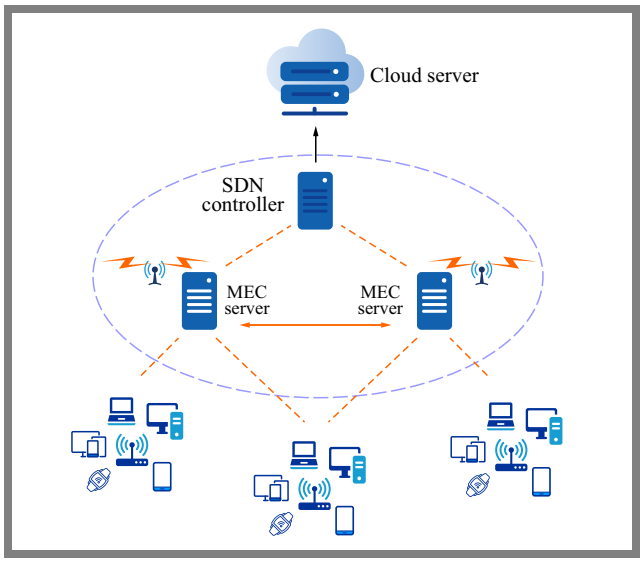

Fig. 1. Structure of an SDN-based MEC environment.

- **Infrastructure layer.** The end-user access layer is the lowest layer of the model. It includes base stations and access points (APs) and supports various user devices: desktops, servers, tablets, mobile phones, smart devices, cars, etc.

In the network model, let $N = \{1, 2, \ldots, n\}$ be the set of mobile devices and $M = \{1, 2, \ldots, m\}$ be the set of edge servers. The SDN controller manages the network topology $G = (V, E)$, where $V = N \cup M$ and E represents the set of communication links.

In the task model, each task T_i is characterized by a tuple (c_i, d_i, τ_i), where c_i is computation requirement (CPU cycles), d_i is data size (bits), and τ_i is the deadline.

Next, in the communication model, the data transmission rate between device i and server j is given by:

$$R_{ij} = B_{ij} \log_2 \frac{P_i \, h_{ij}}{N_0 \, B_{ij}} \,, \tag{1}$$

where B_{ij} is channel bandwidth, P_i stands for transmission power of device i, h_{ij} denotes channel gain, and N_0 is noise power spectral density.

In the computation model, the local execution time for task T_i on device i is:

$$T_{local_i} = \frac{c_i}{f_i} \,, \tag{2}$$

where f_i is the CPU frequency of device i.

The execution time on edge server j is formulated as:

$$T_{edge_{ij}} = \frac{c_i}{f_j} \,, \tag{3}$$

where f_j is the CPU frequency of server j.

For the energy consumption model, the energy consumption for local execution is:

$$E_{local_i} = \kappa \, c_i \, f_i^2 \,, \tag{4}$$

where κ is the energy coefficient.

Energy consumption for offloading is defined as follows:

$$E_{off_{ij}} = P_i \frac{d_i}{R_{ij}} + \varepsilon \, d_i \,, \tag{5}$$

where ε is the coefficient of the circuit power.

The decision variables are defined in the following way:

$$x_{ij} = \begin{cases} 1, & \text{if task } T_i \text{ is offloaded to server} \\ 0 & \text{otherwise} \end{cases} \,, \tag{6}$$

$$y_i = \begin{cases} 1, & \text{if task } T_i \text{ is executed locally} \\ 0 & \text{otherwise} \end{cases} \,. \tag{7}$$

3.1. Problem Formulation

We define the task of offloading and scheduling performed in the MEC-SDN integrated environment as a multi-objective optimization problem. The objective is to reduce the total cost of the system, which is a function of the total task completion time and energy consumption subject to the task deadline constraints.

Let $x_{ij} \in \{0, 1\}$ denote the offloading decision variable, where $x_{ij} = 1$ if task T_i is offloaded to server j, and 0 otherwise. Similarly, let $y_i \in \{0, 1\}$ represent the local execution decision, where $y_i = 1$ if task T_i is executed locally, and 0 otherwise.

The total task completion time (TCT) is given by:

$$TCT = \sum_{i \in N} \left(y_i \, T_{local_i} + \sum_{j \in M} x_{ij} \left(T_{trans_{ij}} + T_{edge_{ij}} \right) \right), \tag{8}$$

where $T_{trans_{ij}} = \dfrac{d_i}{R_{ij}}$ is the transmission time from device i to server j.

Total energy consumption (TEC) is expressed as:

$$TCT = \sum_{i \in N} \left(y_i \, E_{local_i} + \sum_{j \in M} x_{ij} \, E_{off_{ij}} \right) \tag{9}$$

The optimization problem may be formulated as follows:

$$\text{minimize } \; \alpha \cdot TCT + \beta \cdot TEC$$

with the following constraints:

Task allocation:

$$\sum_{j \in M} x_{ij} + y_i = 1, \quad \forall i \in N \tag{10}$$

Task deadline:

$$y_i \, t_{local_i} + \sum_{j \in M} x_{ij} \left(t_{trans_{ij}} + t_{edge_{ij}} \right) \leqslant \tau_i, \quad \forall i \in N \tag{11}$$

Server capacity:

$$\sum_{i \in N} x_{ij} \, c_i \leqslant C_j, \quad \forall j \in M \tag{12}$$

Decision variable:

$$x_{ij}, y_i \in \{0, 1\}, \quad \forall i \in N, \quad \forall j \in M \tag{13}$$

where α and β are time and energy weighting factors, respectively.

Constraint (10) ensures that each task is either offloaded to one of the servers or executed locally. Constraint (11) guarantees that the task completion time does not exceed the deadline.

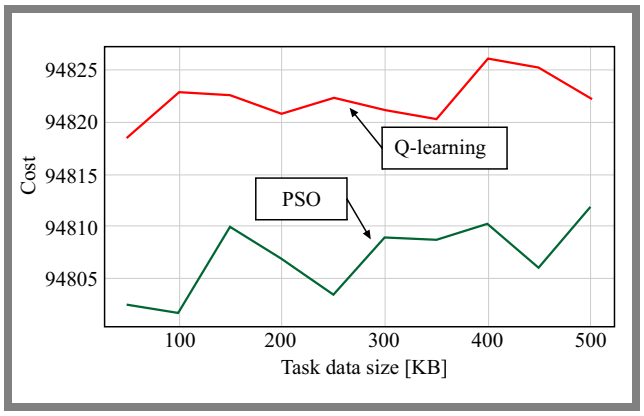

Fig. 2. Offloading cost performance versus task data size.

Constraint (12) ensures that the total computational load on each server does not exceed its capacity.

This formulation emphasizes the details of the task-offloading problem in MEC-SDN environments, considering both time and energy efficiency and respecting system-related constraints [14].

The two methods that have been introduced Algorithm 1 include PSO and Q-learning. The goal is to reduce the total cost of the system, which is a function of the time taken to complete tasks and the energy consumed while running on a dynamic network.

The algorithm starts by setting up the system's variables for the tasks to be performed, the capabilities of the edge server, and the state of the network under control of an SDN controller. Then, PSO is applied, followed by Q-learning, with both methods applied independently to determine their efficiency.

PSO is highly effective at reducing energy consumption during task scheduling, making it particularly suitable for environments with limited resources, where energy efficiency is crucial. In contrast, Q-learning is adept at handling dynamic task loads and showcases excellent scalability as the number of tasks grows.

In the case of PSO, the algorithm adjusts the positions and velocities of particles, reflecting the tasks assigned to servers. It considers the cost of every particle solution, including the local execution and the offloading strategies, which can be full or partial, and then updates the personal best and global best.

For Q-learning, the algorithm gives an agent experience in different episodes. For each episode, it selects actions (servers) for each task from the state, computes rewards according to the cost, and updates the Q-table for better decision-making in the future.

On the same note, another advantage of this algorithm is its versatility in handling both full and partial offloading techniques and its flexibility in adjusting to the dynamic network conditions controlled by the SDN. This makes it possible to assess offloading approaches under different circumstances and conditions.

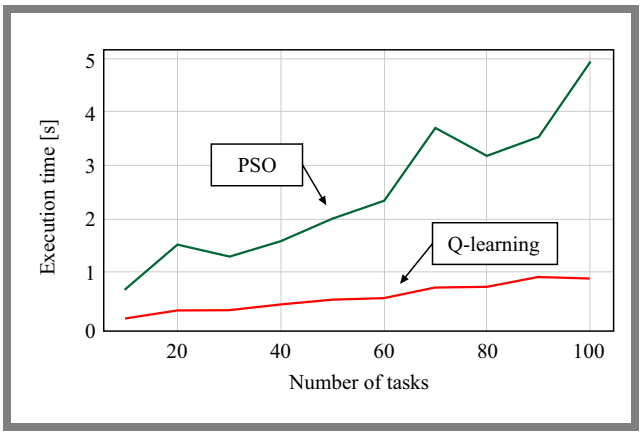

Fig. 3. Scalability test as a function of execution time versus number of tasks.

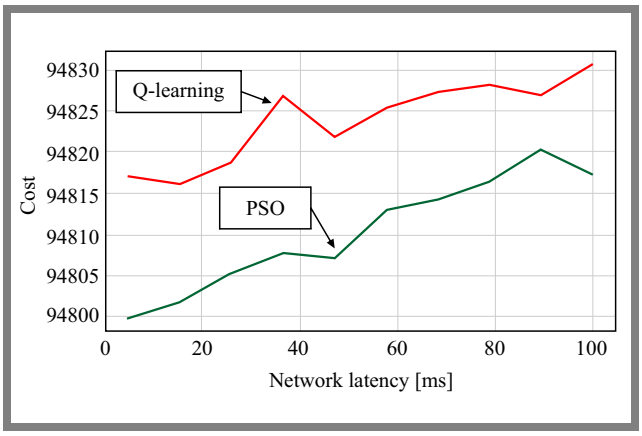

Fig. 4. Effects of network latency on offloading performance.

4. Results and Discussion

Here, simulation results related to PSO-Q are presented and compared with other conventional approaches and advanced algorithms. Such parameters as energy efficiency, task execution time, workload balance, and the capacity for expansion are assessed.

4.1. Offloading Performance and Task Data Size

Figure 2 shows the trends in the costs of PSO and Q-learning, as the task data size increases.

It is observed that both algorithms generate higher costs as the size of the task data increases, which may be attributed to the time taken to transmit the data and the amount of energy consumed. However, the result of PSO is always better than that of Q-learning, in terms of the cost for each learning process in all data sizes.

When data size increases, the difference in performance between PSO and Q-learning grows as well, which indicates that PSO is more suitable for solving large-scale data problems. This could be especially crucial in circumstances where a significant amount of data is required to be transmitted or processed, including multimedia and big data applications relying on edge computing.

Algorithm 1 Energy-efficient task offloading using PSO and Q-learning with SDN

Input: `num_tasks, num_devices, num_servers, task_computation, task_data_size, task_deadlines, server_capabilities, offloading_strategy`

Output: Optimized offloading decisions and system costs for both PSO and Q-learning

```
Initialize SDN_controller, task parameters, and network conditions
Initialize PSO particles and Q-learning agent                          ▷ PSO Optimization
for iteration = 1 to max_iterations do
    SDN_controller.update_network_conditions()
    for each particle do
        total_cost = 0
        for device_id = 1 to num_devices do
            for task_id = 1 to num_tasks do
                server_id = particle.position[task_id]
                local_time, local_energy = local_execution_cost(device_id, task_id)
                if offloading_strategy == full then
                    offloading_time, offloading_energy = full_offloading_cost(device_id,
                    server_id, task_id, SDN_controller)
                else if offloading_strategy == partial then
                    offloading_time, offloading_energy = partial_offloading_cost(device_id,
                    server_id, task_id, SDN_controller)
                end if
                 offloading_cost = offloading_time + offloading_energy
                total_cost += min(local_time + local_energy, offloading_cost)
            end for
        end for
        Update particle's personal best and global best based on total_cost
    end for
    Update particle velocities and positions
end for
                                                                       ▷ Q-learning optimization
for episode = 1 to num_episodes do
    SDN_controller.update_network_conditions()
    total_reward = 0
    for device_id = 1 to num_devices do
        for task_id = 1 to num_tasks do
            state = device_id
            action = Q_agent.choose_action(state)
            local_time, local_energy = local_execution_cost(device_id, task_id)
            if offloading_strategy == full then
                offloading_time, offloading_energy = full_offloading_cost(device_id,
                action, task_id, SDN_controller)
            else if offloading_strategy == partial then
                offloading_time, offloading_energy = partial_offloading_cost(device_id,
                action, task_id, SDN_controller)
            end if
            reward = -min(local_time + local_energy, offloading_time + offloading_energy)
            next_state = (device_id + 1) % num_devices
            Q_agent.learn(state, action, reward, next_state)
            total_reward += reward
        end for
    end for
    Store total_reward for this episode
end for
Return PSO_best_position, PSO_best_cost, Q_Learning_q_table, Q_Learning_rewards
```

When the size of the task data increases from 50 KB to 500 KB, both algorithms demonstrate an increase in costs. As for the cost, PSO clearly shows the lowest result varying from 94800 to 94827. The costs of Q-learning are higher, and they rise from approximately 94819 to 94830. The performance difference is even more pronounced at larger data sizes, and in the case of PSO the cost increase equals approximately 0.003% per kilobyte, compared to 0.002% in the case of Q-learning, and this shows that PSO is more efficient when larger amounts of data are transferred.

4.2. Scalability Test

Figure 3 illustrates the number of tasks and the corresponding execution time of the PSO and Q-learning algorithms. One may notice an interesting balance between the two algorithms. The time it takes PSO to execute its tasks increases at a faster gradient with a growing number of tasks. On the other hand, Q-learning has a comparatively constant execution time, which increases only slightly with the addition of tasks.

Hence, for a small number of tasks, i.e., less than approximately 40, PSO performs better than Q-learning. However, as the number of tasks increases above this point again, Q-learning is more efficient in terms of the time taken to execute the tasks. This crossover point is important for system designers selecting these algorithms.

The execution time of both algorithms escalates from 10 to 100 tasks, as shown in Fig. 3. The execution time of PSO rises more sharply with the number of iterations, from 0 to 5 seconds (614%). Q-learning is more scalable, as evidenced by the fact that the execution time increases from 0.2 to 0. 9 seconds (350%). Q-learning is found to be superior to PSO in terms of execution time for large task numbers.

4.3. Network Latency Effects on Offloading Performance

This graph illustrates the impact of network latency on the offloading costs of the PSO and Q-learning algorithms. As expected, both algorithms demonstrate costs that increase as a function of network latency. This is reasonable, because higher latency would mean that the transmission would take more time to complete, and in some cases the power consumption could be high as well.

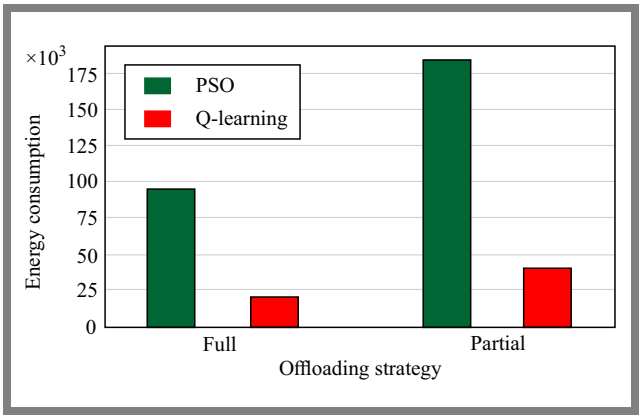

Fig. 5. Energy consumption comparison.

PSO performs better compared to Q-learning at all latency values and has lower costs throughout the range. This implies that PSO might work better in scenarios in which there are fluctuations in network latency in edge computing.

Surprisingly, the difference in performance between PSO and Q-learning is almost constant as latency increases. The fact that both algorithms exhibit parallel growth in costs proves that neither of them is more efficient in high-latency conditions. From the above results, it is clear that as the network latency increases from 0 to 100 ms, both algorithms will generate higher costs. The costs of PSO increase from 94,800 to 94,817 (by 0.018%), while those of Q-learning increase from 94,817 to 94,830 (by 0.014%). The PSO has a relatively lower cost of approximately 13-15 units, irrespective of latency values, proving its better robustness to network delays.

4.4. Energy Consumption Comparison

Figure 5 presents the energy consumption of PSO and Q-learning for full and partial offloading techniques. The findings show that PSO is characterized by better energy utilization. In the full and partial offloading scenarios, PSO uses much less energy than Q-learning. This energy efficiency may be an essential factor in edge computing applications, in which the battery life of the devices is an issue. Surprisingly, partial offloading consumes more power than full offloading for both algorithms. This may seem counterintuitive, as partial offloading is usually used to share the load between local and edge resources. However, this result implies that the cost of splitting tasks and managing partial offloading could be higher than the energy savings it offers.

The main difference between PSO and Q-learning in terms of energy consumption is even more significant in the case of partial offloading of the tasks. This suggests that PSO may be best applied in cases where partial offloading techniques are to be deployed when energy levels are a concern.

The comparison of energy consumption shows that there is a great difference between the full and partial offloading strategies. For full offloading, PSO uses 95,000 units of energy, while Q-learning uses 175,000 units of energy (i.e. 84% more). The results are similar when in the case of partial offloading, where PSO only used approximately 150,000 units, while Q-learning used approximately 185,000 units, (an increase of 23%). This implies that PSO is more energy efficient than the other algorithms, especially in cases where partial offloading is performed.

5. Discussion

The performance metrics shown in Tab. 1 prove that PSO outperforms Q-learning in terms of cost efficiency and energy consumption under all conditions, and especially when the size of data tasks is large. This finding is of significance for practitioners whose responsibilities include optimizing resource allocation in edge computing environments.

Tab. 1. Summary of results.

Metric	PSO	Q-learning	Key observation
Cost (50 – 500 kB of data)	94 800 – 94 827	94 819 – 94 830	PSO is more efficient with larger data sizes
Execution time (10 – 100 tasks)	0.7 – 5 s (614% increase)	0.2 – 0.9 s (350% increase)	Q-learning is more scalable beyond 40 tasks
Cost (0 – 100 ms latency)	94 800 – 94 817	94 817 – 94 830	PSO is more resilient to network delays
Energy (full offloading)	95 000 units	175 000 units	PSO is 84% more energy-efficient
Energy (partial offloading)	150 000 units	185 000 units	PSO is 23% more energy-efficient

For organizations that rely on edge computing for data-intensive applications, multimedia processing, and big data analytics, PSO would save a lot of money. Practitioners are advised to implement PSO in their task-offloading strategies to reduce operational costs without sacrificing performance.

The large difference in energy consumption observed between PSO and Q-learning shows how important energy efficiency is in mobile and edge computing environments. With increasing energy costs and sustainability concerns, organizations will gain from the use of PSO, as it showed an 84% drop in energy usage under full offloading scenarios.

6. Conclusions

This research offers a comparative analysis of the PSO and Q-learning algorithms for task offloading into SDN-integrated MEC scenarios. The approach adopted takes into account full and partial offloading strategies, depending on network conditions controlled by an SDN controller. The results of the simulations show that PSO offers better cost efficiency than Q-learning while handling growing task data sizes and is characterized by lower energy consumption in full and partial offloading.

PSO also demonstrates greater robustness to variations in network latency. Nevertheless, Q-learning shows better scalability than the other methods, and its performance improves when the number of tasks exceeds a specific value. These results may serve as important guidelines for system designers choosing suitable algorithms in based on the requirements of specific MEC scenarios and network conditions.

Future work may focus on examining the integration of the PSO concept with Q-learning with the aim of achieving improved offloading performance. Future work may also consider the influence of more complex network topologies and various types of fog computing resources on offloading performance.

References

[1] M. Satyanarayanan, "The Emergence of Edge Computing", *Computer*, vol. 50, no. 1, pp. 30–39, 2017 (https://doi.org/10.1109/MC.2017.9).

[2] Y. Mao *et al.*, "A Survey on Mobile Edge Computing: The Communication Perspective", *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017 (https://doi.org/10.1109/COMST.2017.2745201).

[3] D. Kreutz *et al.*, "Software-defined Networking: A Comprehensive Survey", *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015 (https://doi.org/10.1109/JPROC.2014.2371999).

[4] Y. Wang *et al.*, "Mobile-edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling", *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268–4282, 2016 (https://doi.org/10.1109/TCOMM.2016.2599530).

[5] H. Guo, J. Liu, and J. Zhang, "Computation Offloading for Multi-access Mobile Edge Computing in Ultra-dense Networks", *IEEE Communications Magazine*, vol. 56, no. 8, pp. 14–19, 2018 (https://doi.org/10.1109/MCOM.2018.1701069).

[6] Y. Wei, F.R. Yu, M. Song, and Z. Han, "Joint Optimization of Caching, Computing, and Radio Resources for Fog-enabled IoT Using Natural Actor-critic Deep Reinforcement Learning", *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2061–2073, 2019 (https://doi.org/10.1109/JIOT.2018.2878435).

[7] L. Yin, J. Luo, and H. Luo, "Tasks Scheduling and Resource Allocation in Fog Computing Based on Containers for Smart Manufacturing", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4712–4721, 2018 (https://doi.org/10.1109/TII.2018.2851241).

[8] Y. Wang *et al.*, "Cooperative Task Offloading in Three-tier Mobile Computing Networks: An ADMM Framework", *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 2763–2776, 2019 (https://doi.org/10.1109/TVT.2019.2892176).

[9] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep Reinforcement Learning Based Computation Offloading and Resource Allocation for MEC", *IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, 2022 (https://doi.org/10.1109/WCNC.2018.8377343).

[10] G. Zhang *et al.*, "Fair Task Offloading Among Fog Nodes in Fog Computing Networks", *IEEE International Conference on Communications (ICC)*, Kansas City, USA, 2018 (https://doi.org/10.1109/ICC.2018.8422316).

[11] L. Tan, Z. Kuang, L. Zhao, and A. Liu, "Energy-Efficient Joint Task Offloading and Resource Allocation in OFDMA-Based Collaborative Edge Computing", in *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1960–1972, 2022 (https://doi.org/10.1109/TWC.2021.3108641).

[12] X. Chen *et al.*, "Multi-tenant Cross-slice Resource Orchestration: A Deep Reinforcement Learning Approach", *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2377–2392, 2022 (https://doi.org/10.1109/JSAC.2019.2933893).

[13] J. Kim *et al.*, "Joint Optimization of Signal Design and Resource Allocation in Wireless D2D Edge Computing", *IEEE INFOCOM 2020 – IEEE Conference on Computer Communications*, Toronto, Canada, 2020 (https://doi.org/10.1109/INFOCOM41043.2020.9155510).

[14] Y. Mao, J. Zhang, S.H. Song, and K.B. Letaief, "Stochastic Joint Radio and Computational Resource Management for Multi-user Mobile-edge Computing Systems", IEEE Transactions on Wireless Communications, vol. 16, no. 9, pp. 5994–6009, 2017 (https://doi.org/10.1109/TWC.2017.2717986).

Fatimah Azeez Rawdhan
Department of Computer Engineering
https://orcid.org/0009-0006-8943-2759
E-mail: fatimah.azeez@uomustansiriyah.edu.iq
Mustansiriyah University, Baghdad, Iraq
https://uomustansiriyah.edu.iq

A Hole-free Shifted Coprime Array for DOA Estimation

Fatimah Abdulnabi Salman[1] and Bayan Mahdi Sabbar[2]

[1]*Al-Nahrain University, Baghdad, Iraq,*
[2]*Al-Mustaqbal University, Baghdad, Iraq*

https://doi.org/10.26636/jtit.2025.1.1959

Abstract — Coprime arrays have recently become a popular trend in estimating the direction of arrival in array signal processing, as they increase the degree of freedom (DOF). Coprime arrays utilize a couple of uniform linear subarrays to create a difference co-array with specific preferable features. In this paper, three proposed structures are considered that depend on the shifting of one of the two uniform linear arrays. The proposed configurations reveal a sequence of lags obtained by filling the holes of the co-array, which span the aperture array. The displacement value depends on the value of the pair of data in the array. The resulting virtual array achieved by means of the proposed methods may generate DOFs *MN* + 1, *MN* + *N* + 1, and *MN* + 2*N* – 2, respectively. The performance of the proposed configurations is evaluated by experimental simulations aiming to demonstrate the effectiveness of the array's design.

Keywords — coprime array, difference co-array, direction of arrival estimation, hole-free array

1. Introduction

Direction of arrival estimation (DOA) is an important topic in array signal processing due to its numerous applications in sonar, wireless communication, radar and navigation [1]–[4] systems. Previously, high-resolution algorithms, such as MUSIC, ESPIRIT, and Root-MUSIC have been proposed to solve the problem of estimating signal direction [5], [6].

These methods are capable of detecting sources with $N-1$ on a uniform linear array with N elements. However, these methods require a large signal-to-noise ratio and numerous snapshots to keep operating properly. In most real cases, the number of snapshots is limited by the operational parameters and physical restrictions that impede efficient DOA estimation [7].

Being able to detect a number of sources that is higher than the number of elements, i.e. increasing the degrees-of-freedom DOFs, has become an interesting topic of research. To cope with this issue, a sparse array structure formed based on a co-array is according to [8], [9], a prospective method capable of increasing DOFs. It attains DOA estimation by locating the sparsest exhibition of the data. A sparse array is composed of two uniform linear subarrays (ULAs).

Distinct sparse arrays, such as the minimum redundancy array (MRA) [10], have been developed to achieve higher DOFs using a limited number of elements. MRA is a sparse array having a maximum number of virtual elements with no holes for the difference co-array, but it lacks general model expression. The DOFs for a certain number of elements cannot be achieved exactly. A nested array [11] may be designed by nesting two ULAs in which the spacing may determine $\mathrm{O}(N2)$ with N elements.

The concept of a co-prime array [12] has attracted the attention of researchers, since co-prime arrays sample the signal sparsely with high resolution and lower cost [2]. The co-prime concept resolves a number of sources that is higher than the number of its elements. Coprime arrays use $M+N-1$ elements to detect $\mathrm{O}(MN)$ sources. Despite a significant number of innovative works relying on the coprime difference co-array, coprime array still suffers from a drawback, as its difference co-array does not provide continuous lags. It has holes that considerably decrease the number of obtainable DOFs.

Several works have focused on proposals to deal with the hole problem. [13] is a good example, where an approach to a coprime array with an extension of one subarray has been proposed. It requires $2M+N-1$ elements to resolve $\mathrm{O}(MN)$ DOFs. The consecutive lags of the difference co-array range are extended with more elements.

The authors of [14] proposed two generalized coprime array configurations with compressed interelement spacing (CACIS) and displaced subarrays (CADiS). For the CACIS configuration, the distance of the elements in the N-subarray is compressed by a compression factor (CF) to keep the minimum distance between the elements, which results in elements overlapping in the self and cross-lag differences. In the CADiS configuration, the N subarray is shifted by a predefined distance to increase the minimum distance between the subarrays in order to expand the aperture size and increase the number of unique lags [14], [15].

However, it breaks the DCA into segments, and critical holes are created that disturb the contiguous lags, which degrades the performance of DOA estimation methods that rely on spatial smoothing.

In article [16], the authors presented extensive research on identifying the location of holes in DCA and proposed two array structures, i.e. the k-times extended coprime array (kECA) and a complementary coprime array (CCA) to fill the holes. In the kECA structure, the M-subarray elements are increased to kM elements to extend the contiguous lags. For the CCA structure, additional elements equal to $M-1$ are

added to fill the holes. The main drawback of the development of these arrays is the extra cost borne due to the presence of additional physical elements.

Furthermore, mutual coupling is affected by the extra element pairs with a small distance present in the kECA structure, and the close element distribution with the distance of half a wavelength in the CCA structure. The authors of [17] proposed a coprime array with suppressed and displaced subarrays (CASDiS) as well as nested displaced coprime subarrays (NesDCoP). For the CASDiS array configuration, the N-subarray is compressed to modify the interelement spacing of the subarray; then, it is shifted by $M + MN$. For the NesDCoP array configuration, the N-subarray is rotated to the negative axis by 180°, compressed by M/CF and then shifted by $N + 1$ to provide hole-free lags.

Article [18] describes a hole-free coprime array (HFCA) developed based on the known number of total elements, in which a maximum number of uDOFs can be achieved by computing the optimal value of M and N.

The problem of holes in coprime arrays may be avoided by redesigning their geometry. In [19], [20] a hole-free array structure is proposed by rearranging the position of the elements in one of the subarrays. The goal is achieved by designing a nested array, with its essentiality property being then analyzed to enhance the array's configuration and extend the aperture array size.

The arrangement of a sparse array in a field affords an adequate but productive manner to plan coprime arrays in order to achieve more lags. Afterwards, only contiguous lags are excluded by means of DOA spatial smoothing estimation techniques, by applying interpolation techniques handling all the lags, or by using sparse array motion. Different coprime configurations, such as generalized coprime and spatial smoothing-MUSIC, have been used in the design process to improve contiguous lags that result in high DOFs.

In this paper, a new array geometry configuration is proposed to improve the available unique lags. A particular emphasis is placed on contiguous lags, and the methods provides a hole-free difference co-array. Thus, spatial efficiency and uniform DOA parameters are exceeded. The array exploits the shifting effects of one coprime array to extend the number of contiguous lags, thus resulting in high DOF. This configuration has resulted in generating hole-free lags.

The paper is structured as follows. Section 2 introduces the fundamental array signal model of coprime arrays, based on difference co-arrays. In Section 3, the proposed array geometry design is presented. The results and conclusions are provided in Sections 4 and 5, respectively.

In this paper, we use upper-case bold characters to represent matrices and lower-case bold characters for vectors. $[\cdot]^T$, $[\cdot]^*$ and $[\cdot]^H$ stand for transpose, the conjugate and conjugate transpose of a vector or matrix, respectively. $\mathrm{Diag}(\cdot)$ and $\mathrm{vec}(\cdot)$ mean a diagonal matrix and the vectorization operator. $\mathrm{E}\{\cdot\}$ represents the expectation operator. I_K indicates a identity matrix with the size of $K \times K$.

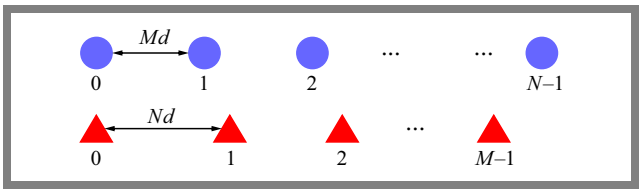

Fig. 1. Configuration of conventional coprime array.

2. Signal Model

The coprime array configuration consists of two ULAs, as illustrated in Fig. 1. N and M are a coprime integer number, such that $N > m$ and $\mathrm{GCD}(M, N) = 1$, where GCD is the largest common divisor, and the two CAs are aligned in a collinear manner. It depends on the concept of the co-array, which refers to the set of points at which the spatial correlation function can be estimated with that array.

The co-array concept has been used in planar design and in estimating the spectrum of multidimensional applications. It may be defined as a set of vectors spacing between points in the elements of given apertures. The vector set is the difference set between the elements or the sum set between array elements of the grid [21]– [23]. The first subarray consists of N sensors with nMd spacing and the second subarray consists of m sensors with mNd spacing. The first element is shared by the two subarrays as a reference element. The total number of sensors that comprise the array is $M + N - 1$.

The elements are positioned at the following locations.

$$\mathbb{P} = \{nMd,\ 0 \leqslant n \leqslant N-1\} \cup \{mNd,\ 0 \leqslant m \leqslant M-1\}\,. \quad (1)$$

$\mathbb{P}$ is a vector that indicates the position of the elements comprising the array $= [p_1, \ldots, p_k]^T$, where $p_i \in \mathbb{P}$, for $i = 1, \ldots, K$.

Considering that D uncorrelated, narrowband, and far-field source signals with power $[\sigma_1^2, \sigma_2^2, \ldots, \sigma_D^2]$, strike the array from angles $\theta = [\theta_1, \theta_2, \ldots, \theta_D]$, the signal received in the array will be declared as:

$$\begin{aligned} x(t) &= \sum_{i}^{D} a(\theta_d)\, S(t) + n(t) \\ &= As(t) + n(t) \in C^{(M+N-1)\times D}\,, \end{aligned} \quad (2)$$

where A is the steering matrix of the style:

$$\begin{aligned} A &= \left[a(\theta_1), a(\theta_2), \ldots, a(\theta_d)\right] \in C^{(M+N-1)\times D} \\ &= \left[1, e^{j\frac{2\pi p_2}{\lambda}\sin(\theta_d)}, \ldots, e^{j\frac{2\pi p_k}{\lambda}\sin(\theta_d)}\right], \end{aligned} \quad (3)$$

$s(t)$ is the signal vector:

$$s(t) = \left[s_1(t), s_2(t), \ldots, s_D(t)\right] \in C^{D\times K}\,, \quad (4)$$

and $n(t)$ is the noise vector:

$$n(t) = \left[n_1(t), n_2(t), \ldots, n_{N+M-1}(t)\right]. \quad (5)$$

The noise is considered independent and distributed randomly with a Gaussian distribution and zero mean invariance. The correlation matrix of the data vector $x(t)$ is collected as:

$$R_{xx} = E\left[x(t)x^H(t)\right] = AR_{SS}A^H = \sigma_n^2 I_{M+N-1}\,, \quad (6)$$

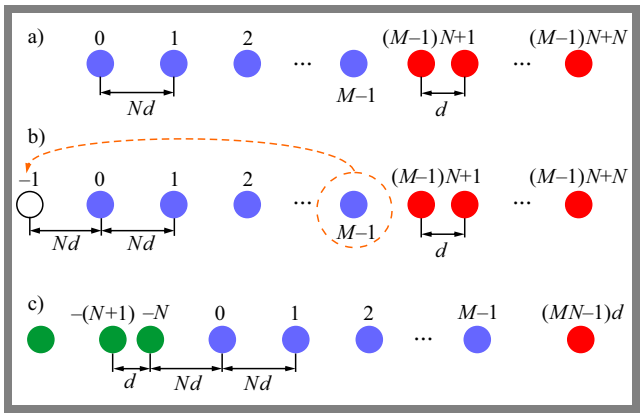

Fig. 2. Hole-free coprime array (FH-CA) configurations for models: a) HF-CA1, b) HF-CA2, and c) HF-CA3.

where $R_{ss} \in C^{D\times D}$ is the source signal covariance matrix and may be expressed as:

$$R_{ss} = \text{diag}\big([\sigma_1^2, \sigma_2^2, \ldots, \sigma_D^2]\big)\,, \in C^{D\times D}. \tag{7}$$

$R_{nn} = \sigma_n^2 I_{M+N-1} \in C^{(M+N-1)\times(M+N-1)}$ is the noise covariance matrix. The covariance matrix is estimated using a limited number of snapshots.

3. Proposed Array Configurations

Three hole-free coprime array (HF-CA) configurations based on PCA on a fixed platform are proposed, as shown in Fig. 2. The first configuration of the HF-CA1 array is illustrated in Fig. 2a. The first subarray consists of M number of elements, while the second subarray consists of N elements. The first subarray elements are positioned at:

$$\mathbb{P}_1 = \big\{0, N, \ldots, (M-1)N\big\}d\,. \tag{8}$$

The second subarray is displaced based on the position of the last sensor in the array, which is $(M-1)N$, and is set as the first sensor location in the new array configuration. The final subarray configuration of the second subarray is as follows:

$$\mathbb{P}_2 = \big\{(M-1)N + n\big\}d, \quad \text{where } 0 \leqslant n \leqslant N\,. \tag{9}$$

The reference element is located at $(M-1)N$, which is shared by the two subarrays, and the total number of the sensor array is $M+N$. The final array geometry for the HF-CA1 array is expressed as:

$$\begin{aligned}\mathbb{P}^{HF-CA1} &= \mathbb{P}_1 \cup \acute{\mathbb{P}}_2 \\ &= \big\{0, N, \ldots, (M-1)N\big\}d \cup \big\{(M-1)N + n\big\}d\,,\end{aligned} \tag{10}$$

where $n = 0, 1, \ldots, N$, the first sensor is located at zero point, and the last sensor is positioned at MN.

$DCA(\mathbb{D})$ of the HF-CA1 array can be illustrated as follows:

$$\begin{aligned}\mathbb{P}^{HF-CA1} &= \big\{(\mathbb{P}_1 - \acute{\mathbb{P}}_2) \cup (\mathbb{P}_1 - \acute{\mathbb{P}}_1) \cup (\mathbb{P}_2 - \acute{\mathbb{P}}_2)\big\} \\ &= \mathbb{D}_{12} \cup \mathbb{D}_{11} \cup \mathbb{D}_{22}\,,\end{aligned} \tag{11}$$

where:

$$\mathbb{D}_{12} = \big\{(mNd - (M-1)Nd + nd), 0 \leqslant m \leqslant M-1, 1 \leqslant n \leqslant N\big\}\,, \tag{12}$$

$$\mathbb{D}_{11} = \big\{mNd,\ \ 0 \leqslant m \leqslant M-1\big\}\,, \tag{13}$$

$$\mathbb{D}_{22} = \big\{\big((M-1)Nd + nd\big) - \big((M-1)Nd + nd\big), 1 \leqslant n \leqslant N\big\}\,. \tag{14}$$

The resulting virtual HF-CA1 array will be a hole-free solution that can be implemented to any M, N pairs. The properties of HF-CA1 can be summarized as follows:

- It contains contiguous lags ranging from $-MN$ to MN, which means that the number of uDOF is $2MN+1$,
- The number of unique lags is $2MN+1$, which is equal to uDOFs, since it is a hole-free array.

To determine the element that has no impact on the DCA, the following relations are considered:

$$\mathbb{D}_{12} \cap \mathbb{D}_{22} = n,\ \ 1 \leqslant n \leqslant N-1\,, \tag{15}$$

$$\mathbb{D}_{12} \cap \mathbb{D}_{11} = mN,\ \ 1 \leqslant m \leqslant M-1\,. \tag{16}$$

From Eq. (15), and considering the relation $\mathbb{D}_{12} \cap \mathbb{D}_{22} \notin \mathbb{P}_1, \acute{\mathbb{P}}_2$, meaning $n \notin \mathbb{P}_1, \acute{\mathbb{P}}_2$, it is easy to observe that the difference of intersection of the self-lags of the difference and cross-lags difference does not represent the position of any element in the actual elements of the matrix. Therefore, it cannot be considered for determining the non-essential element in the array configuration.

From Eq. (16), one may see that the intersecting elements are part of the actual array (M-subarray) that needs to be considered. To determine which element does not contribute to the DCA, the following preposition is presented.

Let

$$\begin{aligned}\mathbb{D}_3 &= \{\acute{\mathbb{P}}_2 - mNd,\ \ 1 \leqslant m \leqslant M-1\} \\ &= (M-1)Nd + nd - mNd,\ \ \text{for } m = M-1, \\ \mathbb{D}_3 &= (M-1)N + n - (M-1)N = n\,.\end{aligned}$$

It has been shown in the previous relation in Eq. (16) that $n \notin \mathbb{P}_1, \acute{\mathbb{P}}_2$, so the element at position $(M-1)N$ does not affect the resulting virtual array. For $m \leqslant M-1, \mathbb{D}_3 \neq n$, so removing any elements from that set may result in a virtual array with holes.

To create an example illustrating the essential importance of the position of elements, M and N are set to 5 and 6, respectively. According to Eqs. (1)–(3), the elements are positioned at $\mathbb{P} = \{0, 6, 12, 18, 24\}d \cup \{25, 26, 27, 28, 29, 30\}d$. Figure 3a shows the configuration of the HF-CA1 array and its difference co-array, where the DCA is a hole-free array, while Fig. 3b-e shows the arrays and their DCAs when one element is removed.

One may notice that the DCA presented in Fig. 3e is similar to the one from Fig. 3a. Therefore, the element at position $(M-1)N = 24$ is not an essential element and does not affect DCA.

After removing the non-essential element from the array configuration, the DCA($\mathbb{D}$) can be expressed in the following manner:

$$\acute{\mathbb{D}} = \acute{\mathbb{D}}_{12} \cup \acute{\mathbb{D}}_{11} \cup \mathbb{D}_{22}\,, \tag{17}$$

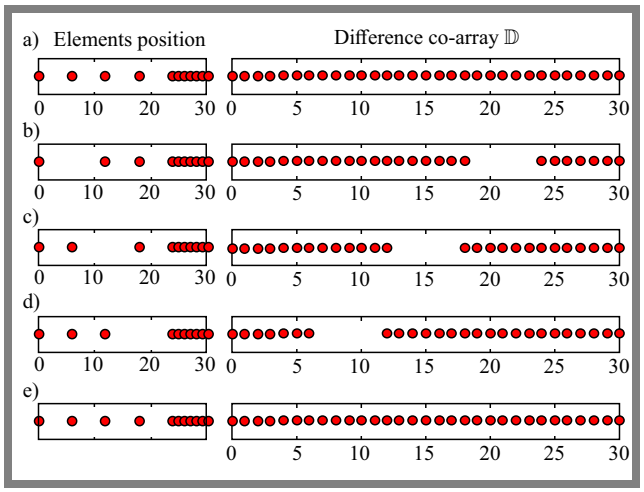

Fig. 3. HF-CA1 configuration and its DCA a), essentiality testing after removing an element from M-subarray at 6 b), 12 c), 18 d), and 24 e), respectively.

where:

$$\begin{aligned}\acute{\mathbb{D}}_{12} &= \big\{(mNd-(M-1)Nd+nd),\\ &\quad 0\leqslant m\leqslant M-2,\ \ 1\leqslant n\leqslant N\big\}\\ \mathbb{D}_{11} &= \big\{mNd,\ \ 0\leqslant m\leqslant M-2\big\}\,.\end{aligned} \tag{18}$$

The configuration of the HF-CA2 array depends on identifying the non-essential element in the HF-CA1 array. The HF-CA2 array is illustrated in Fig. 2b. In this configuration, the nonessential element at position $(M-1)Nd$ is moved to the $-Nd$ location. To justify the new location of the moved elements, the following consideration is presented.

Let us $\delta = \min(\acute{\mathbb{P}}_2) - \max(\acute{\mathbb{P}}_2)$. The reason for selecting $\acute{\mathbb{P}}_2$ is that its self-difference provides a consecutive number with the unit distance between the elements. Regarding the relation given in Eq. (17), the new position of the element may be obtained as follows:

$$\delta = \big\{(M-1)N+\min(n)\big\} - \big\{(M-1)N+\max(n)\big\} = -N\,. \tag{19}$$

The location of the elements in HF-CA2 can be expressed as follows:

$$\mathbb{P}^{HF-CA2} = \mathbb{P}_1 \cup \acute{\mathbb{P}}_2 \cup \mathbb{P}_3\,, \tag{20}$$

where:

$$\mathbb{P}_1 = \big\{mNd,\ \ 0\leqslant m\leqslant (M-1)N\big\}\,, \tag{21}$$

$$\acute{\mathbb{P}}_2 = \big\{(M-1)N+n\big\}d\,, \tag{22}$$

$$\mathbb{P}_3 = -Nd\,. \tag{23}$$

This configuration will reduce the redundancy rate and increase the number of uDOFs. The properties of the HF-CA2 array can be summarized as follows:

- it contains contiguous lags ranging from $MN-N$ to $MN+N$, which means the number of uDOFs is $2MN+2N+1$. The number of uDOFs in the HF-CA2 array is increased by N.
- the number of unique lags is $2MN+2N+1$, since it is a hole-free array, the number of unique lags is equal to the number of uDOFs.

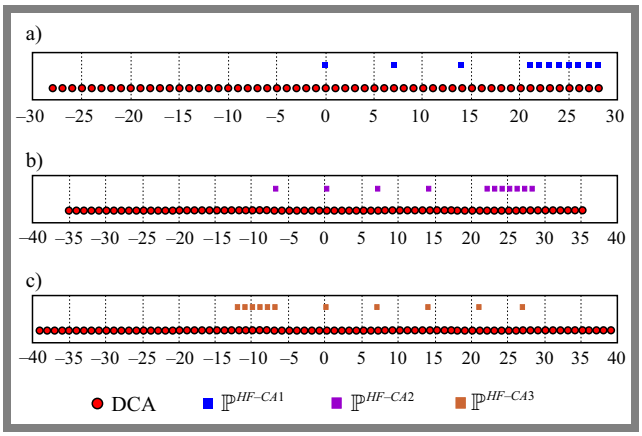

Fig. 4. Example of the HF-CA configuration when $M=4$ and $N=7$ for: a) HF-CA1, b) HF-CA2, and c) HF-CA3.

To achieve more DOFs with particularly contiguous lags, the HF-CA3 configuration is proposed by rotating the dense N-subarray by 180° along the negative side, then the last element in the N-subarray is relocated to the $MN-1$ position, as demonstrated in Fig. 2c. The following relation expresses the position of elements in HF-CA3:

$$\mathbb{P}^{HF-CA3} = \mathbb{P}_1 \cup \mathbb{P}_2 \cup \mathbb{P}_3\,, \tag{24}$$

where:

$$\mathbb{P}_1 = \big\{mNd,\ \ 0\geqslant m\geqslant (M-1)N\big\}\,, \tag{25}$$

$$\mathbb{P}_2 = \big\{-(N:2N-2)\big\}d\,, \tag{26}$$

$$\mathbb{P}_3 = (MN-1)d\,. \tag{27}$$

The idea behind the HF-CA3 configuration is presented below. The subarrays with the location set:
$\mathbb{P}_1$ and $\mathbb{P}_2 = \{-(N:2N-1)\}d$, i.e. the rotated N-subarray, form a hole-free virtual array ranging from $-MN-N+1$ to $MN+N-1$ which is less than the HF-CA2 configuration. If the last element in $\mathbb{P}_2$ is removed, the DCA will have holes located at positions $(mN+N-1)d,\ \ 0\leqslant m\leqslant M-1$. To fill the holes and extend the DCA, the removed element is positioned at $(MN-1)d$, which represents the location of the last element in the hole set.

The properties of the HF-CA3 array can be summarized as follows:

- It contains contiguous lags ranging from $-MN-2N+3$ to $MN+2N-3$, meaning the number of uDOFs is $2MN+4N-5$. The number of uDOFs in the HF-CA3 array is increased by $N-3$.
- The number of unique lags is $2MN+4N-5$, since it is a hole-free array, the number of unique lags is equal to the number of uDOFs.

For illustration, an example is shown in Fig. 4 for the HF-CA configuration with $M=4$ and $N=7$. This is the deployment of the location of the actual elements and the

Tab. 1. Comparison of the closed-form expression for HF-CA1, HF-CA2, and HF-CA3 array configurations with other array types.

Array type	Aperture size	Consecutive lags	Total number of elements
HF-CA1	$MN+1$	$2MN+1$	$M+N$
HF-CA2	$MN+N+1$	$2MN+2N+1$	$M+N$
HF-CA3	$MN+2N-2$	$2MN+4N-5$	$M+N$
Ma-HFCA [18]	$3M+N(K+1-N-\frac{M}{2})-1$	$2N(K+1-N-\frac{N}{2})+6M-1$	K
CADSiS [17]	$2MN+M$	$4MN+1$	$2M+N$
NesDCoP [17]	$2MN+N$	$4MN+2N+1$	$2M+N$
CCA [16]	$kMN-N$	$2kMN-2N+1$	$(k+1)M+N-2$
k-times [16]	$kMN-N$	$2(k-1)MN+2M-1$	$kM+N-1$
ECA [13]	$2MN-N$	$2MN+2M-1$	$2M+N-1$

DCA of the three HF-CA configurations. One may notice that all the HF-CA configurations are hole-free arrays. HF-CA1 is capable of generating a ULA segment within the $[-28:0:28]$ range, while HF-CA2 and HF-CA3 may generate ULA segments within the $[-35:0:35]$ and $[-39:0:39]$ ranges, respectively. HF-CA3 can achieve the largest uDOF with an extension to the array aperture size, which makes it capable of identifying more sources.

Table 1 shows the comparison of the closed form expressions of the lags generated by HF-CA1, HF-CA2, and HF-CA3 array configurations, with different coprime array types, depending on the array aperture size, contiguous lags and the total number of elements.

4. Simulation Results

The proposed HF-CA designs were tested using Matlab to verify the weight function, array robustness, spatial spectrum, and RMSE. The weight function $w(m)$ of the M,N pair, $m \in D$ is the number of elements pairs that have the same value in the DCA index m. The weight function $w(m)$ of the ULA having M,N elements meets the following characteristics [15]:

$$w(0)=(M,N), \sum_{m\in D} w(m)=(M,N)^2, w(m)=w(-m)\ . \tag{28}$$

The weight function gives an indication of the element allocation in an array. The weight functions $w(m)$ show different distributions of the virtual array elements. A smaller weight function means that there are fewer pairs with one partition that is who getting a sparse array structure. As the weight function is minimized, the root means square error (RMSE) is decreased as well [24].

Figure 5 illustrates the weight function, worked out according to Eq. (28), of the proposed array designs with different array types, such as PCA, ECA, CACIS, CADiS, k-times ECA, CCA, CASDiS, NesDCoP, and Ma-HFCA with 9 elements. In the figure, the blue dots represent the positions of the physical elements, while the red dots represent the virtual lags and the red crosses represent the locations of holes. One may notice that PCA, ECA, k-times ECA, CACIS, and CADiS cannot provide a hole-free sparse array and the range for their ULA segment is $[-9:0:9]$, $[-14:0:14]$, $[-14:0:14]$, $[-16:0:16]$, and $[8:23,-8:-23]$, respectively.

The remaining array structures provide a hole-free co-array with an ULA segment. The consecutive sets are: $[-15:0:15]$, $[-20:0:20]$, $[-25:0:25]$, $[-23:0:23]$, $[-20:0:20]$, $[-25:0:25]$ and $[-27:0:27]$ for CCA, CASDiS, NesDCoP, Ma-HFCA, HF-CA1, HF-CA2, and HF-CA3, respectively.

It can be seen that the proposed HF-CA3 outperforms its rivals, as it has the highest number of consecutive lags compared to other array types. Another remark regarding the ULA segments for CASDiS, HF-CA1 and NesDCoP, HF-CA2 for 9 elements is that these array designs can have the same ranges of consecutive lags. This is not always true for other numbers of the elements, as can be seen from Tab. 2.

Considering the weight functions related to Eq. (28), as shown in Fig. 5, $w(1)$ for CADiS is zero, since there is no element in the first position (there is a hole). $w(1)=2$ for PCA, ECA, k-times ECA and CASIC. $w(1)=3$ for CCA, Ma-HFCA and HF-CA3. $w(1)=4$ for CADiS, NesDCoP and HF-CA2. $w(1)=5$ for CASDiS and HF-CA2, while $w(2)=2$ for PCA, ECA, k-times ECA, Ma-HFCA and HF-CA3. $w(2)=3$ for NesDCoP and HF-CA2. $w(2)=4$ for CADiS, CASDiS and HF-CA1, and $w(2)=5$ for CCA and CACIS.

4.1. Evaluation of Robustness

Various array structures are evaluated based on their resistance to failure. In the exercise, the location of the antenna may lead to some disturbance, including the antenna's failure in some radical situations. Several parameters are used to evaluate robustness, including spatial efficiency and redundancy rate.

Spatial efficiency is the ratio between the number of contiguous lags and the length of the virtual array aperture for the positive side in a sparse array [25].

Tab. 2. Comparison of different array types against aperture, number of DOFs, and operational robustness.

Array type	M, N pairs	Number of elements	Array aperture	Number of uDOFs	Spatial efficiency	Redundancy rate
ECA [13]	(3, 4)	9	20	29	69.23%	56.79%
k-times [16]	$K = 2$, (3, 4)	9	20	29	69.23%	56.79%
CCA [16]	$K = 2$, (2, 5)	9	15	31	100%	66.12%
NA [9]	(4, 5)	9	25	51	100%	39.50%
CACIS [14]	(4, 5)	9	21	33	79.48%	54.32%
CADiS [14]	(4, 5)	9	27	16	74.20%	44.44%
CASDiS [17]	(2, 5)	9	19	39	100%	61.00%
NesDCoP [17]	(2,5)	9	25	51	100%	37.04%
Ma-HFCA [18]	(2, 3)	9	23	47	100%	42.00%
HF-CA1	(4, 5)	9	20	41	100%	49.40%
HF-CA2	(4, 5)	9	25	51	100%	37.04%
HF-CA3	(4, 5)	9	27	55	100%	32.10%
ECA [13]	(3, 8)	13	40	53	64.55%	60.35%
k-times [16]	(6, 7)	13	48	97	100%	42.60%
CCA [16]	(5, 9)	13	40	53	79.48%	54.32%
NA [9]	(6, 7)	13	50	30	60%	44.44%
CACIS [14]	$K = 3$, (3, 4)	13	40	65	79.74%	56.80%
CADiS [14]	$K = 2$, (3, 5)	13	25	51	100%	64.58%
CASDiS [17]	(4, 5)	13	44	81	90.80%	49.70%
NesDCoP [17]	(4, 5)	13	45	91	100%	41.42%
Ma-HFCA [18]	(3, 5)	13	48	97	100%	42.60%
HF-CA1	(6, 7)	13	42	85	100%	49.70%
HF-CA2	(6, 7)	13	49	99	100%	41.42%
HF-CA3	(6, 7)	13	53	107	100%	36.68%

$$\eta = \frac{\text{Number of uDoFs}}{\text{Array aperture size}} . \tag{29}$$

Spatial efficiency has an impact on signal determination and estimation efficiency. Higher spatial efficiency of the coprime virtual array structure may ensure high DOA estimation performance, meaning fewer waste elements in the DCA.

Table 2 shows the spatial efficiency for five different types of arrays. It can be noted that the proposed HF-CA array configuration ensures 100% spatial efficiency, when compared with the remaining types. The HF-CA array designs provide contiguous lag with a hole-free array configuration, which is equal to the unique lags.

The lagged redundancy rate is the measure of repeated spatial lag for pairs of elements in an array. It can be defined as [16]:

$$r_{redun} = \frac{|\mathbb{P}|^2 - |\mathbb{D}|}{|\mathbb{P}|^2} \tag{30}$$

where set $\mathbb{P}$ represents the positions of the physical array elements and set $\mathbb{D}$ stands for DCA.

Although the redundancy rate can reverberate robustness to some extent, it suffers from some constrictions. When redundancy is limited to specific levels, high levels of robustness cannot be obtained even though there is a high redundancy rate. From Fig. 5, one may notice that the highest level of redundancy is centered on the zeroth element position, and it is equal to the number of the physical elements in the sparse array structure. It can be noted that CADiS and the second proposed array configuration have no positions without any redundancy.

The DOA estimates of spatial spectrum of the proposed array structures are shown in Fig. 6.

The array configuration is based on $M = 4, N = 5$, so the total number of elements in the array is 9. The simulation parameters are set to 10 dB, the number of snapshots is 500, and the source angle is θ_i uniformly distributed within the range $-60°, \ldots, 60°$. For the HF-CA1 structure, the number of sources is set to $Q = 18$. The design of the HF-CA1 matrix can generate 41 DOFs within the $[20 : 0 : 20]$ range. From Fig. 6a, one may notice that the HF-CA1 array can estimate

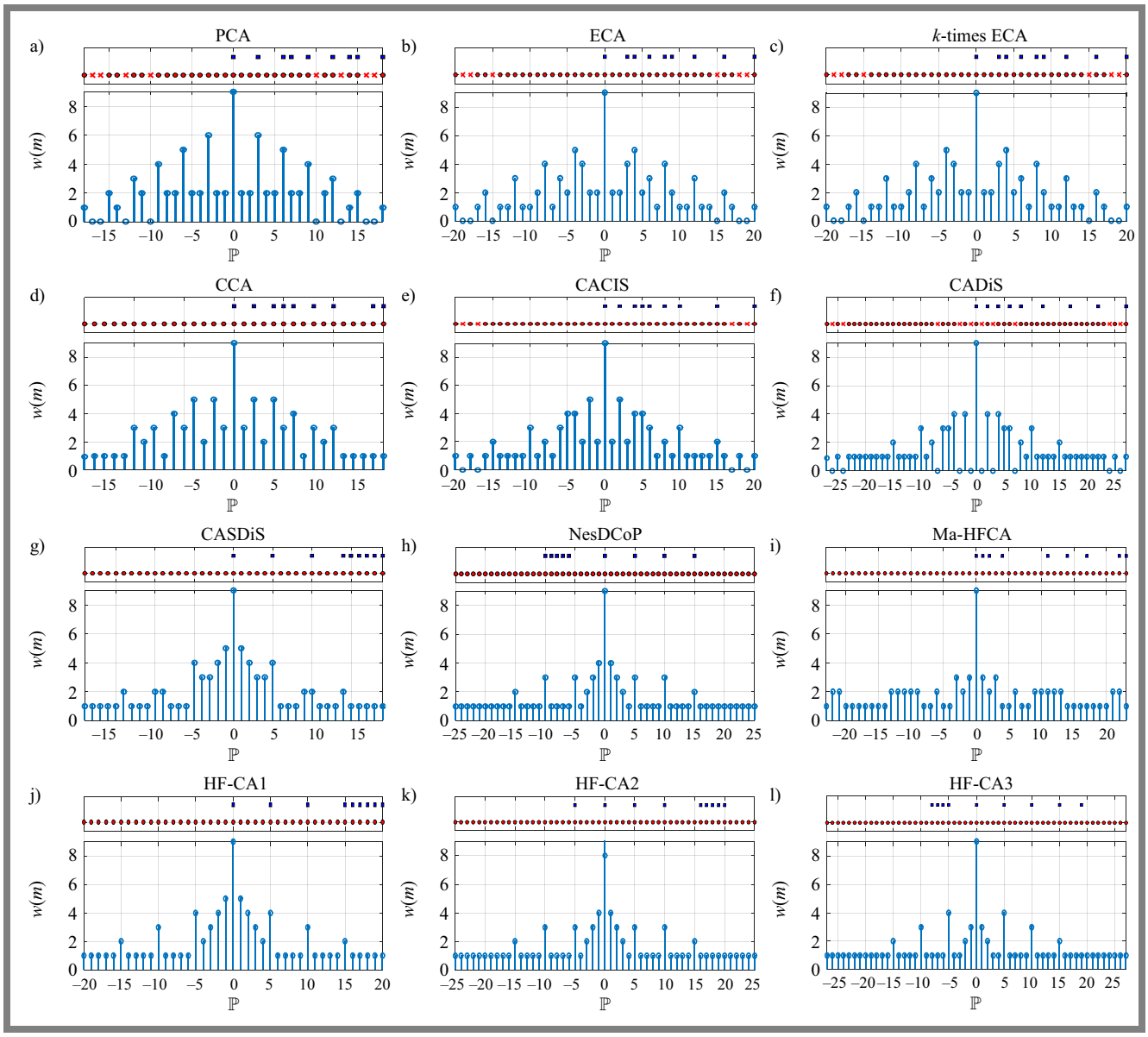

Fig. 5. Weight function for different array configurations.

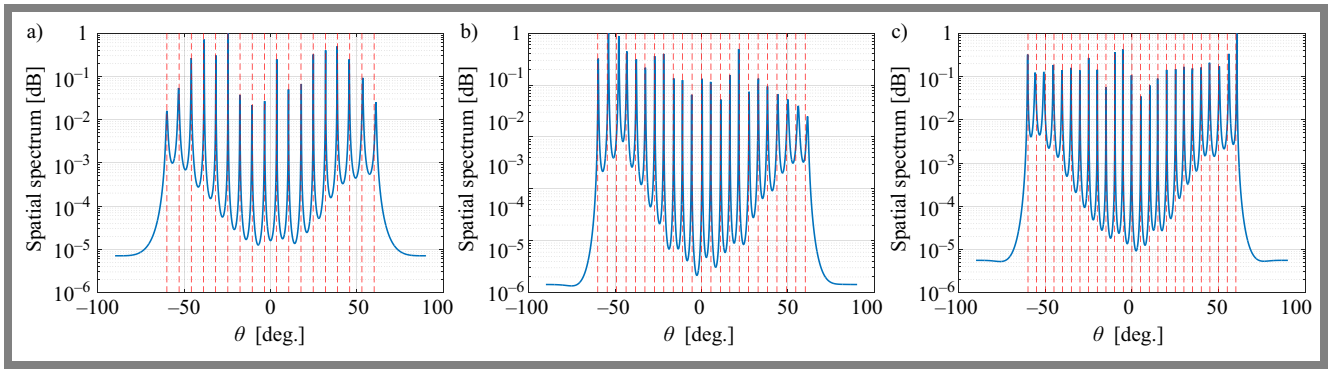

Fig. 6. Spatial spectrum estimation ($\{textSNR = 10\ dB$ and snapshot $= 500$) for: a) HF-CA1 with $Q = 18$, b) HF-CA2 with $Q = 23$, and HF-CA3 with $Q = 25$.

18 sources effectively. For the HF-CA2 structure, the number of sources is set to $Q = 23$. The HF-CA2 array design can generate 51 DOFs within the $[-25 : 0 : 25]$ range. It can resolve the 23 sources accurately, as shown in Fig. 6b. For HF-CA3, the number of sources is set to $Q = 25$ and all the sources can be resolved correctly as shown in Fig. 6c,

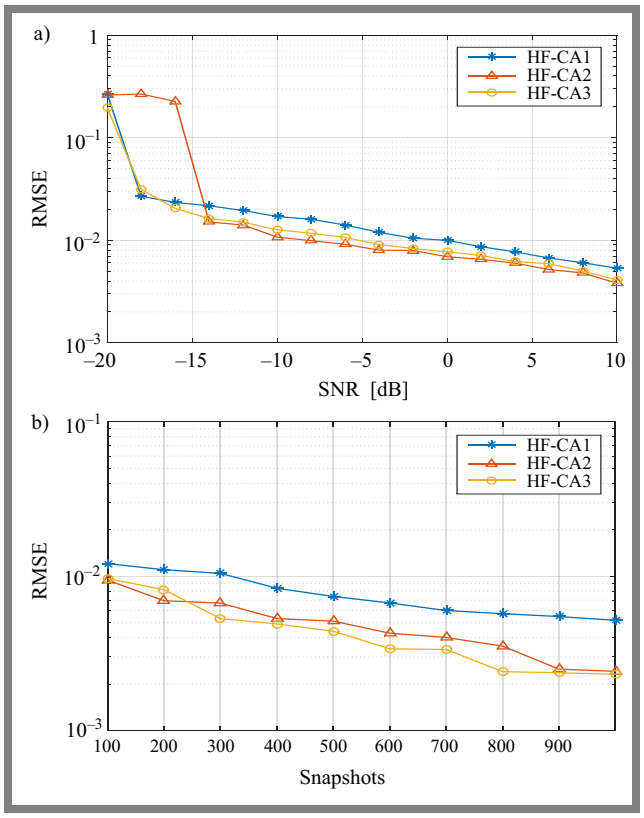

Fig. 7. RMSE evaluation for HF-CA for: a) snapshots $= 500$ and $D = 12$ sources and b) SNR $= 0$ and $D = 12$ sources.

since HF-CA3 can generate a uniform segment within the $[-27 : 0 : 27]$ range and the number of DOFs is 55.

4.2. RMSE Evaluation

The root mean square error (RMSE) is one of the most common metrics that is used to evaluate the accuracy of DOA estimation. Calculations of the error between the true and estimated DOA are given in [26]–[27]:

$$RMSE = \sqrt{\frac{1}{QM_C}\sum_{i=1}^{M_C}\sum_{q=1}^{Q}(\tilde{\theta}_{q(i)} - \theta_q)^2}\,, \qquad (31)$$

where M_C denotes the number of total Monte Carlo trials, Q is the number of sources and $\tilde{\theta}_{q(i)}$ are the true and estimated DOA, respectively.

The RMSE vs. SNR ratio is shown in Fig. 7a. It can be seen that HF-CA2 and HF-CA3 configurations outperform the HF-CA1 array configuration, because there are more DOFs used for DOA estimation. Figure 7b illustrates the RMSE vs. snapshots ratio. HF-CA2 and HF-CA3 methods outperform the HF-CA1 configuration as the number of snapshots increases.

5. Conclusions

In this paper, hole-free coprime array structures are proposed with the positions of their elements being shifted and moved from one location to another. These proposed methods can achieve a higher number of contiguous lags and a hole-free array structure, when compared with other structures. The performance of the array structure was evaluated using different robustness parameters, such as spatial efficiency and redundancy rate, in addition to aperture size and number of DOFs, and a comparison with other array structures was performed. The results of simulations and numerical analyses revealed the effectiveness of the proposed methods.

References

[1] H. Krim and M. Viberg, "Two Decades of Array Signal Processing Research: The Parametric Approach", *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–64, 1996 (https://doi.org/10.1109/79.526899).

[2] Z. Weng and P.M. Djuric, "A Search-free DOA Estimation Algorithm for Coprime Arrays", *Digital Signal Processing*, vol. 24, pp. 27–33, 2014 (https://doi.org/10.1016/j.dsp.2013.10.005).

[3] F.A. Salman and B.M. Sabbar, "DOA Estimation Exploiting Moving Platform of Unfolded Coprime Array", *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 2, pp. 532–542, 2022 (https://doi.org/10.22266/ijies2022.0430.47).

[4] F.A. Salman and B.M. Sabbar, "Estimation of Coherent Signal on Modified Coprime Array", *16th International Middle Eastern Simulation and Modelling Conference*, 2020.

[5] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation", *IEEE Transaction Antenna Propagation*, vol. 34, no. 3, pp. 276–280, 1986 (https://doi.org/10.1109/TAP.1986.1143830).

[6] R. Roy and T. Kailath, "ESPRIT – Estimation of Signal Parameters via Rotational Invariance Technique", *IEEE Transactions Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989 (https://doi.org/10.1109/29.32276).

[7] W. Baxter and E. Aboutanios, "Fast Direction of Arrival Estimation in Coprime Arrays", *International Conference on Radar (RADAR)*, Brisbane, Australia, 2018 (https://doi.org/10.1109/RADAR.2018.8557304).

[8] X. Wang, X. Wang, and X. Lin, "Co-prime Array Processing with Sum and Difference Co-array", *49th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, USA, 2015 (https://doi.org/10.1109/ACSSC.2015.7421152).

[9] X. Wang, Z. Chen, S. Ren, and S. Cao, "DOA Estimation Based on the Difference and Sum Co-array for Coprime Arrays", *Digital Signal Processing*, vol. 69, pp. 22–31, 2017 (https://doi.org/10.1016/j.dsp.2017.06.013).

[10] A.T. Moffet, "Minimum-redundancy Linear Arrays", *IEEE Transactions on Antennas and Propagation*, vol. 16, no. 2, pp. 172–175, 1968 (https://doi.org/10.1109/TAP.1968.1139138).

[11] P. Pal and P.P. Vaidyanathan, "Nested Arrays: A Novel Approach to Array Processing with Enhanced Degrees of Freedom", *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, 2010 (https://doi.org/10.1109/TSP.2010.2049264).

[12] P.P. Vaidyanathan and P. Pal, "Sparse Sensing with Co-prime Samplers and Arrays", *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573–586, 2011 (https://doi.org/10.1109/TSP.2010.2089682).

[13] P. Pal and P.P. Vaidyanathan, "Coprime Sampling and the Music Algorithm", *Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, Sedona, USA, 2011 (https://doi.org/10.1109/DSP-SPE.2011.5739227).

[14] S. Qin, Y.D. Zhang, and M.D. Amin, "Generalized Coprime Array Configuration for Direction of Arrival Estimation", *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1377–1390, 2015 (https://doi.org/10.1109/TSP.2015.2393838).

[15] A. Raza, W. Liu, and Q. Shen, "Thinned Coprime Arrays for DOA Estimation", *25th European Signal Processing Conference (EUSIPCO)*, Kos, Greece, 2017 (https://doi.org/10.23919/EUSIPCO.2017.8081236).

[16] X. Wang and X. Wang, "Hole Identification and Filling in K-times Extended Co-prime Arrays for Highly-efficient DOA Estimation", *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2693–2706, 2019 (https://doi.org/10.1109/TSP.2019.2899292).

[17] K. Shabir, T.H. Al Mahmud, R. Zheng, and Z. Ye, "Generalized Super-resolution DOA Estimation Array Configurations' Design Exploiting Sparsity in Coprime Arrays", *Circuits Systems and Signal Processing*, vol. 38, pp. 4723–4738, 2019 (https://doi.org/10.1007/s00034-019-01078-1).

[18] P. Ma, J. Li, F. Xu, and X. Zhang, "Hole-free Coprime Array for DOA Estimation: Augmented Uniform Co-array", *IEEE Signal Processing Letters*, vol. 28, pp. 36–40, 2021 (https://doi.org/10.1109/LSP.2020.3044019)

[19] F.A. Salman and B.M. Sabbar, "Semi-symmetrical Coprime Linear Array with Reduced Mutual Coupling Effect and High Degrees of Freedom", *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1483–1495, 2024 (https://iiict.uob.edu.bh/IJCDS/papers/IJCDS1501105_1570871127.pdf).

[20] F.A. Salman and B.M. Sabbar, "Initial Phase Effect on Direction Finding Using Coprime Array", *International Middle Eastern Simulation and Modelling Conference*, Baghdad, Iraq, 2022.

[21] D.A. Linebarger, I.H. Sudborough, and I.G. Tollis, "Difference Bases and Sparse Sensor Arrays", *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 716–721, 1993 (https://doi.org/10.1109/18.212309).

[22] J.H. McClellan, "Multidimensional Spectral Estimation", *Proceedings of the IEEE*, vol. 70, no. 9, pp. 1029–1039, 1982 (https://doi.org/10.1109/PROC.1982.12431).

[23] S.W. Lang and J.H. McClellan, "Spectral Estimation for Sensor Arrays", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 2, pp. 349–358, 1983 (https://doi.org/10.1109/TASSP.1983.1164080).

[24] C.-L. Liu and P.P. Vaidyanathan, "Super Nested Arrays: Linear Sparse Arrays with Reduced Mutual Coupling – part 1: Fundamentals", *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3997–4012, 2016 (https://doi.org/10.1109/TSP.2016.2558159).

[25] Y.D. Zhang, S. Qin, and M.G. Amin, "DOA Estimation Exploiting Coprime Arrays with a Sparse Sensor Spacing", *IEEE International Conference on Acoustic, Speech, and Signal Processing*, Florence, Italy, 2014 (https://doi.org/10.1109/ICASSP.2014.6854003).

[26] G. Qin, M.G. Amin, and Y.D. Zhang, "DOA estimation exploiting sparse array motions", *IEEE Transactions on Signal Processing*, vol. 67, no. 11, pp. 3013–3027, 2019 (https://doi.org/10.1109/TSP.2019.2911261).

[27] F.A. Salman and B.M. Sabbar, "Low-complexity DOA Estimation Method Based on Joined Coprime Array", *Journal of Telecommunications and Information Technology*, no. 1, pp. 11–16, 2024 (https://doi.org/10.26636/jtit.2024.1.1350).

Fatimah Abdulnabi Salman, Ph.D.
System Engineering Department
https://orcid.org/0000-0001-8875-9844
E-mail: faty_salman@nahrainuniv.edu.iq
Al-Nahrain University, Baghdad, Iraq
https://nahrainuniv.edu.iq

Bayan Mahdi Sabbar, Prof.
Medical Instrumentation Techniques Engineering Department
https://orcid.org/0000-0003-0541-2410
E-mail: prof.dr.bayan.mahdi@uomus.edu.iq
Al-Mustaqbal University, Baghdad, Iraq
https://uomus.edu.iq

Context-Awareness for Device-to-Device Resource Allocation

Marcin Rodziewicz

Poznan University of Technology, Poznań, Poland

https://doi.org/10.26636/jtit.2025.1.1934

Abstract — The paper investigates a context-aware approach to radio resource allocation for device-to-device (D2D) communication, focusing on solutions that leverage information on user equipment location and environmental features, such as building layouts. A system enabling direct communication by sharing uplink resources with cellular users is considered. Such a system introduces mutual interference between direct and cellular communications, posing challenges related to maintaining adequate performance levels. To address these challenges, various context-based resource allocation methods are analyzed, aiming to optimize spectral efficiency and minimize interference. The study explores the impact that different D2D device densities exert on overall network performance measured by means of spectral efficiency and the signal-to-interference ratio.

Keywords — cellular network, context-awareness, device-to-device, resource allocation

1. Introduction

Novel system concepts are being explored to address the growing demand for mobile data traffic in cellular networks.

Among these, device-to-device (D2D) communication, which enables direct wireless links between pieces of user equipment (UE), is particularly promising. Unlike conventional cellular connections that route traffic through the base stations (BSs) and the core network, D2D communication allows UE to communicate directly when the individual devices are close to each other, leading to higher data rates, lower energy consumption, and reduced transmission delays.

D2D communication is expected to help offload traffic from future radio access networks (RAN) and support a wide range of new services, including vehicle-to-everything communication. Such applications, however, introduce new design challenges for future systems, particularly related to ensuring strict quality of service (QoS) and reliability for applications that may involve large numbers of active users.

D2D communication is often envisioned to work as underlay to cellular networks, meaning that it shares the radio resources with the primary system. By allowing D2D devices to share the same radio resources with cellular users, direct communication may potentially push the frequency reuse factor (FRF) beyond one. However, this spectrum sharing poses challenges, particularly around managing interference, which calls for advanced power control and resource allocation mechanisms to maintain network performance.

The main focus of this paper is on resource allocation methods for D2D communications that leverage context-awareness in their operation.

The remainder of the paper is organized as follows. In Section 2, a short review of existing works is presented. Section 3 contains the description of the system model under consideration. Section 4 presents the proposed resource allocation solutions. Section 5 evaluates the considered approaches and discusses the results achieved, while Section 6 presents the conclusions.

2. Related Works

Many D2D-related studies ([1]–[7], and [8]) are focused on mitigating interference in D2D communications. The most commonly used approaches include power control and resource allocation solutions. For example, in [6], a D2D power reduction method was suggested to control interference with cellular users. In [9], a location-based power control mechanism was proposed to enhance the parameters of D2D communications. With regards to resource allocation solutions, a review of the literature focusing on this aspect, and D2D in general, may be found in [10]. Many solutions presented in the survey utilize slowly varying channel parameters, such as path loss or shadowing for resource allocation and D2D management, with paper [2] being one of the examples here.

A newer survey [11] showcases solutions utilizing artificial intelligence (AI) for resource allocation. It lists several data-driven machine learning (ML) approaches that could be used to enhance resource allocation in D2D communication networks. These approaches leverage the ability of ML models to learn complex patterns and make real-time decisions. For example, the authors in [3] present a weighted cooperative Q-learning-based resource selection (WCopQLRS) strategy. Unlike independent learning schemes, WCopQLRS incorporates weighted Q-value exchanges among D2D pairs within a defined cooperation range to minimize interference and optimize energy efficiency. This approach improves system throughput, energy consumption, and fairness by leveraging cooperative learning among neighboring D2D pairs.

Another paper utilizing ML for resource allocation is [12]. The authors developed a multi-agent deep Q-network (DQN) framework to optimize mode selection and channel allocation in heterogeneous cellular networks. This model maximizes

the system sum rate while satisfying the QoS requirements of cellular and D2D users. Each D2D agent operates independently with partial information sharing, thus reducing system complexity. The proposed approach is claimed to achieve higher sum rates and QoS satisfaction rates while converging faster than the baseline methods under consideration. Additionally, its distributed architecture ensures scalability and robustness in heterogeneous environments. These are just two examples from the vast set of references included in [11]. The number of publications exploring the use of ML in the context of D2D communication shows that this is an area worth more attention in the future.

However, as mentioned in the introduction, this paper focuses on context-aware resource allocation methods for D2D communications, where resource allocation strategies leverage contextual information such as the location of users, with some works exploring the possibility of using this information for that specific purpose [9], [13]–[18]. For example, in [13] and [15] a power control mechanism and an interference control strategy using an interference limited area (ILA) constraint were proposed. The users located in this area were excluded from the resource sharing scheme. The purpose of the resource sharing area constraint is to ensure that the probability of a D2D communication outage caused by interference from cellular users is lower than a predefined threshold value.

In [18], a resource-sharing criterion with distance limitation was introduced to reduce the set of cellular users who can share resources with D2D users, resulting in a reduced probability of D2D link outage. An additional advantage of the solution proposed in [18] is that it does not require cellular users to reduce their transmission power, thus avoiding degradation in cellular link quality. Another paper utilizing context-awareness is [19]. Coverage performance in D2D networks, which is the main topic of the paper, has received less attention compared to throughput and energy efficiency studies.

The authors of [19] address this by constructing a cluster-based UE classification and spectrum-sharing allocation model for multi-tier hybrid heterogeneous networks. The presented approach classifies UEs into clusters based on their locations, distinguishing between cluster center and edge UEs. By analyzing the coverage probability of these clusters, the scheme dynamically adjusts spectrum sharing to enhance resource utilization and minimize interference. This location-aware classification ensures that edge devices, which typically face more interference, receive appropriate resource allocations.

D2D communication is often considered in the context of vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communications. In [21], a resource allocation scheme for D2D communication based on channel measurements is presented, ensuring proportional fairness among users while maximizing the overall throughput of the system. The proposed method uses allocation in long time slots to improve the system's efficiency and fairness. In [22], a resource allocation method for vehicle-to-everything (V2X) communication scenarios based on D2D is introduced, taking into account the realistic assumption of imperfect channel state information (CSI). The proposed algorithm aims to maximize the ergodic capacity of the user devices in the vehicle while meeting quality of service requirements.

However, to the best of the author's knowledge, not many studies consider using contextual information for V2V resource allocation, with [23]–[25] and [26] serving as good examples here. The authors of [26] propose a location-based approach to V2V communications, leveraging location stability to enhance energy efficiency for both cellular and vehicular users by reducing computational demands. A location-partition-based channel allocation and power control method for C-V2X networks is presented in [25], dividing the coverage area into zones to simplify resource allocation and improve latency while minimizing interference in high-density scenarios.

Additionally, [24] examines the robustness of location-based D2D resource allocation schemes against positioning errors, finding that these methods generally maintain strong performance despite inaccuracies in position estimates, with only minimal impact on throughput. Finally, [23] compares location-based and CSI-based methods for resource allocation in D2D-enabled networks, showing that while CSI-based approaches offer higher spectral efficiency, they also require significant feedback, especially in dynamic environments.

In this paper, methods utilizing context-awareness, in the form of device location and building layout information, in the radio resource allocation mechanism for D2D communication, are presented. Unlike the works listed above, which considered a single cell system, the proposed mechanism is evaluated in a multi-cell environment with an FRF of 1.

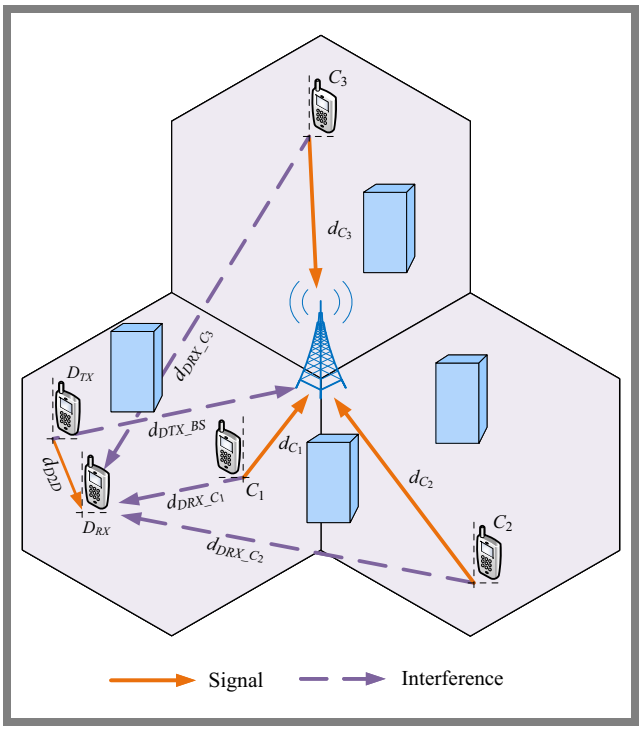

Fig. 1. System model [9].

3. System Description

In this paper, a multi-cell cellular system employing orthogonal frequency division multiple access (OFDMA) with a frequency reuse factor (FRF) of 1 is considered. D2D communication underlay is enabled by sharing uplink (UL) resources with cellular users (CUE, cellular user equipment). An illustrative diagram is presented in Fig. 1. It is worth mentioning that the considered system is generic and is not directly related to a particular cellular system standard. However, the solutions presented are applicable to widely used OFDMA-based systems, including LTE-A and 5G.

Sharing the uplink radio resources leads to the introduction of additional interference in the system for both cellular and direct communication. Interference affecting the receiving D2D device (DUE, D2D user equipment) occurs when cellular users transmit on the same radio resources. Conversely, from the perspective of CUE, interference caused by transmitting DUE is experienced by the serving base station. One way of mitigating this interference, as mentioned in Section 2, is to use a proper radio resource allocation method. The signal-to-interference ratio (SIR) for the DUE receiver γ_D and the base station for the k-th cellular user γ_{C_k} are given by:

$$\gamma_D = \frac{h_D(d)\, P_D}{\sum\limits_{i=1}^{N} h_{DC_i}(d) P_{C_i}} \tag{1}$$

and

$$\gamma_{C_k} = \frac{h_{C_k}(d)\, P_{C_k}}{\sum\limits_{i=1, i\neq k}^{N} h_{C_i}(d) P_{C_i} + h_D(d) P_D}\,, \tag{2}$$

where N is the number of neighboring cells using the same frequency, including the cell where the D2D pair is located.

The parameters $h_D(d)$ and $h_{DC_i}(d)$ represent the distance-dependent path losses between the D2D users and between the DUE receiver and the i-th CUE transmitter, respectively. $h_{C_k}(d)$, $h_{C_i}(d)$, and $h_D(d)$ represent path losses between the k-th and i-th cellular transmitters and the base station, as well as between the DUE transmitter and the base station, respectively. P_D is the transmit power of the DUE, and P_{C_i} is the signal power transmitted by the i-th CUE transmitter. In the considered system, an open loop power control (OLPC) mechanism is used to set the transmission power:

$$P_C = \min\left(P_0 + A\, h(d), P_{max}\right)\,, \tag{3}$$

where P_0 is the initial power level of the device, A is the path loss compensation factor, and $h(d)$ is the path loss between the transmitter and the receiver. The maximum transmission power is limited by P_{max}.

In developing the resource management method for D2D communication, several assumptions were made. First, the goal of the proposed solution is to minimize the impact of cellular traffic on direct communications. This assumption is based on the fact that the base station possesses greater processing capabilities, enabling the deployment of advanced mechanisms to reduce interference from direct communications.

The second assumption is a centralized management approach, meaning that a D2D control node is introduced into the system. This node is associated with a set of base stations serving a specific area and has knowledge of the locations of the devices it serves, as well as the layout of buildings in the area covered. The allocation of resources for D2D devices is implemented on top of the allocation of resources for CUE devices.

4. Proposed Solutions

In this study, two approaches to the resource allocation problem were considered, with both of them aiming to minimize interference at the D2D receiver. The first approach works by measuring the links between devices which are later reported to the control node. This solution requires knowledge of the channel state between all nodes in the system, not just between the users and the base station, which results in a significant signaling overhead. Therefore, this solution is impractical, but it serves as a reference point for the second approach in which contextual information is used for resource management purposes.

Two context-aware methods are considered: the first one relies solely on information concerning the location of users in the cellular network, while the other one uses not only location, but also knowledge of the layout of buildings in the covered area. All the mechanisms mentioned use the same resource allocation procedure, differing only in how resource-sharing

Algorithm 1 Find sharing candidates – Location

Input: Set of D2D pairs
Output: Lists of sharing candidates

for each BS attached to D2D control node **do**
  for each CUE scheduled for transmission **do**
    if $d_{DTX\_BS} > d_{d2d}$ **and** $d_{CTX\_DRX} > 0.5 \cdot d_{CTX\_BS}$ **then**
      Add CUE to list of sharing candidates
    end if
  end for
  Sort the list with d_{CTX_DRX} in descending order
end for

Algorithm 2 Find sharing candidates – Map

Input: Set of D2D pairs
Output: Lists of sharing candidates

for each BS attached to D2D control node **do**
  for each CUE scheduled for transmission **do**
    Evaluate line-of-sight conditions for the D2D receiver and the sharing candidate
    if $d_{DTX\_BS} > d_{d2d}$ **and** is NLoS **then**
      Add CUE to list of sharing candidates
    end if
  end for
  Sort the list with d_{CTX_DRX} in descending order
end for

candidates are selected from the set of CUEs scheduled for transmission at specific times.

For each transmission time interval (TTI), resources are first allocated to cellular users at each base station connected to the D2D control node. Then, a set of D2D pairs is selected according to the round robin algorithm. The size of this set depends on the length of the allocation (i.e. how many resource blocks are available for UEs) for CUE users at the base stations. Subsequently, for each D2D pair, a sorted list of candidates for resource sharing is created for each base station. Depending on the chosen approach, as described below, this list is created in a different way.

A method relying on the location of users (Location). In this method, the positions of UE pieces are known. Based on that information, distances between them are calculated. The procedure of finding the candidates is presented in Algorithm 1. The goal of this method is to maximize the distance between the sharing devices, as this potentially minimizes interference. Four distances are considered: distance between the D2D devices (d_{d2d}), distance from the D2D transmitter to the base station (d_{DTX_BS}), distance from the CUE sharing candidate to the D2D receiver (d_{CTX_DRX}), and distance from this candidate to its serving base station (d_{CTX_BS}).

For each scheduled CUE from each base station, these distances are evaluated. If the distance between the D2D transmitter and the base station is less than the D2D distance, and the distance from the sharing candidate to the D2D receiver is more than half of its distance to its respective base station, the candidate is added to the list. Once all candidates meeting these criteria are identified, the list is sorted, in descending order, according to the distance between each candidate and the D2D receiver.

A method relying on location and building layout (Map). This method is an extension of the location-based approach, where in addition to the information concerning the location of users, knowledge of the layout of buildings in the area under consideration is used. The procedure is presented in Algorithm 2. In this case, the condition for adding a given CUE to the list of resource-sharing candidates is further restricted by visibility conditions. The method assumes that only candidates without a direct line-of-sight are added to the list. Similarly to the location-based method, after considering all candidates, the resulting list is sorted in descending order based on the distance between the CUE candidate and the D2D receiver.

A method using channel state reporting (Min-int). Presented in Algorithm 3, this method assumes that path losses and expected transmit power levels are known. Based on this information, the level of interference between devices is determined, specifically the interference from the CUE sharing candidate to the D2D receiver is taken into consideration (I_{CTX_DRX}). The calculated interference level is used to sort the list of resource-sharing candidates in ascending order.

The lists generated using the methods described above are used in the next step of the allocation procedure, i.e. in the selection of specific CUE devices for the D2D pairs considered in the allocation round. The order of processing the

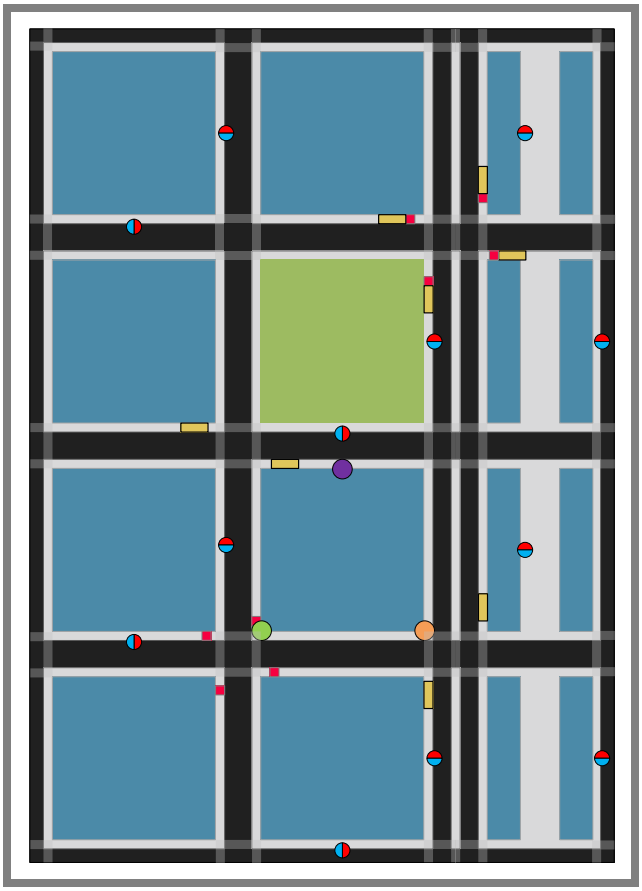

Fig. 2. Simulation deployment environment: Madrid grid model. Green, purple and orange dots represent the antennas of the considered macro BS.

D2D pairs considered in the round is determined based on the distance between the devices forming a given pair. Resources are first assigned to pairs with the largest distance between devices. This is because the probability of low SIR at the receiver in such pairs, due to greater path loss between the pair, is higher than for pairs with a small distance between devices. It may happen that the same sharing candidate is selected for multiple D2D pairs. In this case, a simple conflict resolution mechanism is employed which involves selecting the next candidate from the list. If all candidates on the list have been

Algorithm 3 Find sharing candidates – Min-int

Input: Set of D2D pairs
Output: Lists of sharing candidates

```
for each BS attached to D2D control node do
    for each CUE scheduled for transmission do
        Based on known channel loss values and
        transmitted signal powers
        Calculate the interference from the D2D
        transmitter to the BS
        Calculate the interference from the CUE
        candidate to D2D receiver
        Add CUE to list of sharing candidates
    end for
    Sort the list with I_CTX_DRX in ascending order
end for
```

Tab. 1. Considered scenarios.

	Scenario 1	Scenario 2	Scenario 3
Pedestrian CUEs	320	335	305
In-vehicle CUEs	320	345	295
Pedestrian DUEs	60	46	76
In-vehicle DUEs	100	74	124

Tab. 2. Network spectral efficiency for different scenarios and resource allocation methods. All values are in [bps/Hz].

Scenario	Method	DL spectral efficiency	UL spectral efficiency
1	Location	3.97	2.98
	Map	4.05	3.07
	Min-int	4.14	3.04
	No D2D	2.54	1.61
2	Location	4.09	3.09
	Map	4.16	3.16
	Min-int	4.42	3.32
	No D2D	2.49	1.58
3	Location	4.02	3.02
	Map	4.08	3.10
	Min-int	4.19	3.07
	No D2D	2.53	1.58

considered, but not selected, the D2D pair is excluded from the current resource allocation round.

Once all CUE candidates are assigned to the D2D pairs considered in the round, the procedure's final step is to adjust the initial CUE transmission plan so that each pair of candidates shares the same resources.

5. Simulation and Results

5.1. Simulation Environment

The resource allocation solutions proposed in Section 4 were investigated using system-level simulations of an OFDMA-based cellular network with a frequency reuse factor of one. The simulation tool used in the experiments was co-developed by the author and implemented according to the guidelines set in the METIS project [27]. More details about the tool can be found in [28].

The simulation tool implements channel models defined by METIS [29]. These models, unlike the more commonly used ones, employ 3D map-based real-time methods to assess line-of-sight conditions between the nodes. METIS channel models are used for cellular users. For D2D users, a modified version of the D2D model defined by ITU-R [30] is applied. The modification involves using a map, instead of a statistical approach as defined in the ITU-R recommendation, to determine the visibility conditions between specific devices.

The study considers a cellular network deployed in an urban environment according to the Madrid grid model (MGM) (Fig. 2) [27]. The MGM deployment includes 12 micro base stations and a single macro base station and covers an area of 387 by 552 meters. The MGM incorporates essential environmental characteristics, such as building heights and detailed street layouts typical of a European city [27]. These elements are vital for reliably evaluation of signal propagation and interference, and thus offer a good point of departure for assessing resource allocation models.

The considered cellular network consists of a macro base station placed on top of the tallest building at a height of 50 m. The base station operates in the frequency division duplex (FDD) mode in three sectors with antennas placed on the edge of a building, as shown in Fig. 2. Each sector has a directional antenna with a pattern defined in [27]. The azimuths of sector antennas are 0°, 120°, and 240°, relative to the north. Each sector is operating on a 2.6 GHz carrier frequency using 80 MHz of bandwidth. The round robin method was used to allocate resources to CUE users, assigning successive resource blocks to each CUE device in turn.

In the simulation environment, 800 users were evenly distributed outdoors, either on sidewalks or in vehicles. These users were further divided into 4 groups:

– pedestrian CUE devices,
– in-vehicle CUE devices,
– pedestrian DUE devices,
– in-vehicle DUE devices.

Different configurations of these groups were considered in the investigations. The goal was to examine the impact of the number of D2D devices on network performance, assuming the use of the considered resource allocation methods. The configurations analyzed are grouped into three scenarios, as summarized in the Tab. 1. Scenario 1 considered 80 D2D pairs (160 UEs) with the distance between each device in a D2D pair randomly drawn from a range of 0 to 50 m, according to a uniform distribution, taking into account that the distance between users in cars is constrained by the assumed dimensions of the vehicle. Out of all the pairs in Scenario 1, 30 were pedestrian users, while the remaining 50 D2D pairs were placed in vehicles.

In Scenario 2, the allowable number of D2D pairs was reduced from 80 to 60 (a 25% reduction), while in Scenario 3, this number was increased by 25% to 100 pairs. In all cases, the maximum distance between D2D devices was 50 meters, and the ratio of pedestrians to in-vehicle users was approximately 0.6.

The mobility of the users (including vehicles) is also modeled according to the METIS guidelines. The simulations were repeated 50×, and each simulation lasted 10 s.

Various system performance statistics were gathered during the simulations, with the most important of them being:

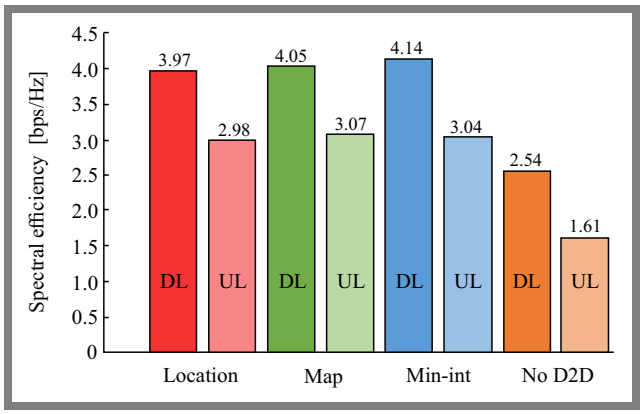

Fig. 3. Network spectral efficiency for Scenario 1.

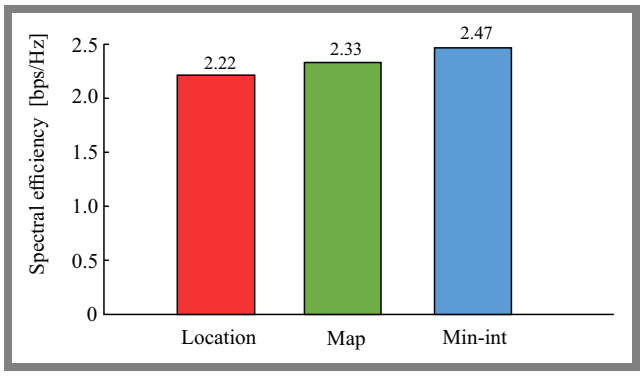

Fig. 4. D2D spectral efficiency for Scenario 1.

– spectral efficiency (expressed in bits/s/Hz) for downlink (DL) and uplink (UL) (Fig. 3 and Tab. 2), and for active D2D users (Fig. 4 and Tab. 3),
– cumulative distribution function (CDF) (Fig. 5 and Fig. 6) and related metrics (Tab. 4 and Tab. 5) of the signal-to-interference ratio for base stations and D2D communication receivers.

5.2. Results

The first set of results (Fig. 3) presents the overall spectral efficiency of the system for the downlink (left bar) and uplink (right bar) for each of the proposed resource allocation methods in Scenario 1. Additionally, for reference, simulation results where direct communication was not allowed (denoted as No D2D) are included. When analyzing the graph and focusing on the downlink results, one may notice that the reference measurement-based method (min-int) achieves the best results. The map-based method (map) is the runner up, followed by the location-based (location) approach. However, it is worth noting that the differences between them amount to several percentage points only. When analyzing the uplink results, we see that the differences are even smaller, with the map-based method performing the best. When comparing all the results with a system without any direct communications, the benefits of introducing D2D communication become clearly visible. The noticeable increase in the system's spectral efficiency is achieved by offloading the core network and raising the frequency reuse factor beyond one.

Tab. 3. D2D spectral efficiency for different scenarios and resource allocation methods. All values are in [bps/Hz].

Scenario	Method	Spectral efficiency
1	Location	2.22
	Map	2.33
	Min-int	2.47
2	Location	2.43
	Map	2.53
	Min-int	2.92
3	Location	2.19
	Map	2.29
	Min-int	2.44

The spectral efficiency of the network was determined for all the considered deployment scenarios. The results are summarized in Tab. 2. The conclusions from comparing allocation methods in each scenario are the same as above, but one can notice a certain difference between the scenarios. We can observe that an increased number of D2D pairs does not automatically improve the system's overall efficiency. In this case, a reduction of the number of devices, in relation to Scenario 1, has led to a greater efficiency gain. While increasing the number of pairs also improved efficiency compared to Scenario 1, the improvement was smaller than in Scenario 2. This suggests that there may be an optimal number of D2D pairs in a network that maximizes its overall spectral efficiency.

The simulations also provided results regarding the spectral efficiency of the devices capable of forming D2D pairs (Fig. 4). In this graph, a trend that is similar to the one visible in Fig. 3, may be observed. However, in this case, the differences between different allocation methods are larger, reaching a maximum of approx. 12%. As before, the location-based method achieves the lowest effectiveness. However, the incorporation of maps may enhance its performance. The method relying on channel state measurements delivers the best results, as it possesses the most precise knowledge of transmission conditions. However, as mentioned earlier,

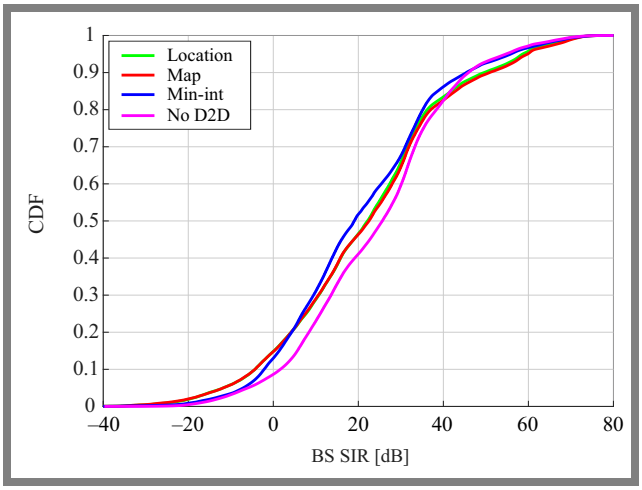

Fig. 5. SIR at the base station in Scenario 1.

Tab. 4. Median and 10th percentile SIR at the BS for different scenarios and resource allocation methods. All values are in [dB].

Scenario	Method	Median	10th perc.
1	Location	22.141	–4.3065
	Map	22.499	–4.3058
	Min-int	19.041	–2.1492
	No D2D	25.631	1.6646
2	Location	21.223	–4.9543
	Map	21.474	–4.6288
	Min-int	18.077	–2.5884
	No D2D	25.122	0.8034
3	Location	22.798	–3.7181
	Map	23.152	–3.5979
	Min-int	19.677	–1.5049
	No D2D	25.098	1.5184

conducting measurements and reporting the channel state between all devices in the network would result in excessive signaling overhead. By comparing D2D spectral efficiency in different scenarios (Tab. 3), one may notice that an increase in the number of D2D pairs in the system leads to more competition for resources between D2D pairs, with the overall D2D communication performance suffering as a result. A possible solution to this problem could be a more sophisticated scheduling algorithm than the round robin approach used in the simulations.

The next investigated aspect was the impact of D2D communication on the signal-to-interference ratio at the base station. Fig. 5 shows the cumulative distribution function of SIR at the base station in Scenario 1. The differences between the considered cases are not very pronounced, with the median SIR equaling 22.1, 22.5, 19, and 25.6 dB for the Location, Map, Min-int and No D2D cases, respectively. It can be observed that the min-int method has the greatest impact on the base station's SIR. This is mainly due to the lack of additional constraints, such as distance or line-of-sight, when

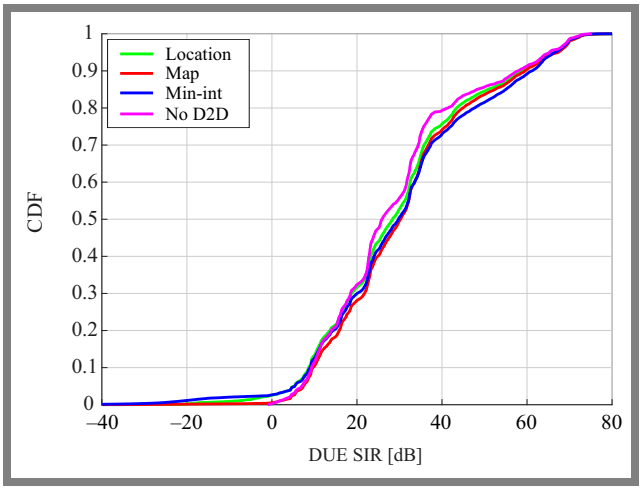

Fig. 6. SIR at the D2D receiver in Scenario 1

Tab. 5. Median and 10th percentile SIR at the D2D receiver for different scenarios and resource allocation methods. All values are in [dB].

Scenario	Method	Median	10th perc.
1	Location	28.360	8.6160
	Map	30.249	8.6160
	Min-int	29.740	8.9721
	No D2D	25.887	9.2703
2	Location	27.750	8.9115
	Map	29.745	9.7949
	Min-int	30.466	9.3058
	No D2D	25.628	8.8109
3	Location	28.939	9.3148
	Map	30.559	10.7750
	Min-int	30.365	9.6873
	No D2D	25.550	9.3708

adding devices to the list of sharing candidates. We also see that the location-based method and the map-based approach exert very similar impacts on the base station's performance.

When analyzing the base station's SIR statistics for the considered deployment scenarios, as presented in Tab. 4, we find that a decrease in the number of devices (Scenario 2) results in a lower median SIR compared to the other cases. However, the differences in the median values are very small and we should also consider the reference SIR value for the No D2D case in each scenario in the comparison. More noticeable differences are visible in the 10th percentile statistic, but the trend according to which a lower number of D2D devices exerts a higher impact on SIR at the base station is still upheld.

SIR of the D2D pair devices may be analyzed in a similar manner (Fig. 6). It can be noted that all three methods protect D2D communication to a similar degree, with the location-based method slightly underperforming compared to the others, in the range from the median to the 90th percentile. For example, the difference in the median compared to the map-based method is approximately 2 dB (30.25 dB for the map-based method and 28.36 dB for the location-based method). This difference is a result of the additional protection imposed by the visibility conditions in the map-based method.

Such an approach increases the SIR but does not necessarily mean better system performance, as this restriction may result in fewer transmission opportunities. Looking at Tab. 5 in which the median and the 10th percentile statistics for D2D SIR in Scenarios 1–3 are presented, one may notice that once again a reduction in the number of D2D devices lowers the median SIR. However, the same cannot be said about the 10th percentile SIR. In this case, both an increase and a reduction in the number of D2D devices in relation to Scenario 1 lead to a higher value of this statistic.

When analyzing both spectral efficiency and SIR, it is worth considering why a reduction in the number of D2D pairs results in better spectral efficiency, despite worse SIR statistics.

This is likely due to the increased number of transmission opportunities for D2D pairs, as reduced competition allows for achieving higher spectral efficiency in this scenario, compared to other approaches.

6. Conclusions

The paper presents and analyzes resource allocation methods utilizing contextual information, such as the location of users and buildings layout. The context-aware methods are compared with each other and with a reference method that operates using measurements and channel state reporting. The study shows that context-aware methods may be used effectively to support resource allocation in direct communications.

Analysis of the impact that D2D device density exerts on the system's performance, performed in the course of this study, indicated that an excessive number of D2D devices can negatively affect overall performance measured by means of spectral efficiency.

Additionally, it is demonstrated that introducing direct communication in a cellular system brings several benefits, such as increased spectral and energy efficiency (due to transmissions over smaller distances). It was also shown that the impact of D2D communication on the performance of a cellular system turned out to be minimal in the scenarios under consideration.

In future work, a comparison with ML-based allocation methods could be conducted to further evaluate the proposed context-aware resource allocation solutions. Such a comparison would provide valuable insights into the performance trade-offs between pure context-based approaches and ML models.

References

[1] J. Gu, S.J. Bae, B.-G. Choi, and M.Y. Chung, "Dynamic Power Control Mechanism for Interference Coordination of Device-to-Device Communication in Cellular Networks", *2011 Third Internat. Conference on Ubiquitous and Future Networks (ICUFN)*, Dalian, China, 2011 (https://doi.org/10.1109/icufn.2011.5949138).

[2] P. Janis *et al.*, "Interference-aware Resource Allocation for Device-to-Device Radio Underlaying ellular Networks", *VTC Spring 2009 – IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, 2009 (https://doi.org/10.1109/VETECS.2009.5073611).

[3] S. Sharma and B. Singh, "Weighted cooperative reinforcement learning-based energy-efficient autonomous resource selection strategy for underlay D2D communication", *IET Communications*, vol. 13, no. 14, pp. 2078–2087, 2019 (https://doi.org/10.1049/iet-com.2018.6028).

[4] N. Reider and G. Fodor, "A Distributed Power Control and Mode Selection Algorithm for D2D Communications", *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, art. no. 266, 2012 (https://doi.org/10.1186/1687-1499-2012-266).

[5] J. Seppala, T. Koskela, T. Chen, and S. Hakola, "Network Controlled Device-to-Device (D2D) and Cluster Multicast Concept for LTE and LTE-A Networks", *2011 IEEE Wireless Communications and Networking Conference*, Cancun, Mexico, 2011 (https://doi.org/10.1109/wcnc.2011.5779270).

[6] C.-H. Yu, O. Tirkkonen, K. Doppler, and C. Ribeiro, "On the Performance of Device-to-Device Underlay Communication with Simple Power Control", *VTC Spring 2009 – IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, 2009 (https://doi.org/10.1109/vetecs.2009.5073734).

[7] C.-H. Yu, O. Tirkkonen, K. Doppler, and C. Ribeiro, "Power Optimization of Device-to-Device Communication Underlaying Cellular Communication", *2009 IEEE International Conference on Communications*, Dresden, Germany, 2009 (https://doi.org/10.1109/icc.2009.5199353).

[8] M. Zulhasnine, C. Huang, and A. Srinivasan, "Efficient Resource Allocation for Device-to-Device Communication Underlaying LTE Network", *2010 IEEE 6th Int. Conference on Wireless and Mobile Computing, Networking and Communications*, Niagara Falls, Canada, 2010 (https://doi.org/10.1109/wimob.2010.5645039).

[9] M. Rodziewicz, "Location-based Power Control Mechanism for D2D Communication Underlaying a Cellular System", *Journal of Telecommunications and Information Technology*, vol. 3, pp. 49–53, 2023 (https://doi.org/10.26636/jtit.2023.3.1361).

[10] T. Islam and C. Kwon, "Survey on the State-of-the-art in Device-to-Device Communication: A Resource Allocation Perspective", *Ad Hoc Networks*, vol. 136, art. no. 102978, 2022 (https://doi.org/10.1016/j.adhoc.2022.102978).

[11] T. Rathod and S. Tanwar, "AI-based Resource Allocation Techniques in D2D Communication: Open Issues and Future Directions", *Physical Communication*, vol. 66, art. no. 102423, 2024 (https://doi.org/10.1016/j.phycom.2024.102423).

[12] Y. Zhi *et al.*, "Deep Reinforcement Learning-based Resource Allocation for D2D Communications in Heterogeneous Cellular Networks", *Digital Communications and Networks*, vol. 8, no. 5, pp. 834–842, 2022 (https://doi.org/10.1016/j.dcan.2021.09.013).

[13] P. Bao and G. Yu, "An Interference Management Strategy for Device-to-Device Underlaying Cellular Networks with Partial Location Information", *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications – (PIMRC)*, Sydney, Australia, 2012 (https://doi.org/10.1109/pimrc.2012.6362830).

[14] X. Chen *et al.*, "Downlink Resource Allocation for Device-to-Device Communication Underlaying Cellular Networks", *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications – (PIMRC)*, Sydney, Australia, 2012 (https://doi.org/10.1109/pimrc.2012.6362746).

[15] H. Min, J. Lee, S. Park, and D. Hong, "Capacity Enhancement Using an Interference Limited Area for Device-to-Device Uplink Underlaying Cellular Networks", *IEEE Transactions on Wireless Communications*, vol. 10, no. 12, pp. 3995–4000, 2011 (https://doi.org/10.1109/twc.2011.100611.101684).

[16] M. Rodziewicz, "Location-based Mode Selection and Resource Allocation in Cellular Networks with D2D Underlay", *European Wireless 2015 – 21th European Wireless Conference*, Budapest, Hungary, 2015 (https://ieeexplore.ieee.org/document/7147698).

[17] M. Rodziewicz, "Wykorzystanie Informacji o Rozmieszczeniu Budynków do Zarządzania Zasobami Komunikacji Bezpośredniej", *Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne*, vol. 1, no. 4, pp. 343–347, 2024 (https://doi.org/10.15199/59.2024.4.77) (in Polish).

[18] H. Wang and X. Chu, "Distance-constrained Resource-sharing Criteria for Device-to-Device Communications Underlaying Cellular Networks", *Electronics Letters*, vol. 48, no. 9, p. 528, 2012 (https://doi.org/10.1049/el.2012.0451).

[19] Y. Lv, X. Jia, C. Niu and N. Wan, "D2D Network Coverage Analysis Based on Cluster User Equipment Classification and Spectrum Sharing Allocation", *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2020 (https://doi.org/10.1109/ICCC51575.2020.9345149).

[20] K. Doppler *et al.*, "Device-to-Device Communication as an Underlay to LTE-advanced Networks", *IEEE Communications Magazine*, vol. 47, no. 12, pp. 42–49, 2009 (https://doi.org/10.1109/mcom.2009.5350367).

[21] X. Li *et al.*, "Resource Allocation for Underlay D2D Communication with Proportional Fairness", *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6244–6258, 2018 (https://doi.org/10.1109/tvt.2018.2817613).

[22] X. Li, L. Ma, Y. Xu, and R. Shankaran, "Resource Allocation for D2D-based V2X Communication with Imperfect CSI", *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3545–3558, 2020 (https://doi.org/10.1109/jiot.2020.2973267).

[23] M. Botsov, S. Stanczak, and P. Fertl, "Comparison of Location-Based and CSI-Based Resource Allocation in D2D-enabled Cellular Networks", *2015 IEEE International Conference on Communications (ICC),* London, UK, 2015 (https://doi.org/10.1109/icc.2015.7248705).

[24] N.P. Kuruvatti *et al.*, "Robustness of Location Based D2D Resource Allocation against Positioning Errors", *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, Glasgow, UK, 2015 (https://doi.org/10.1109/vtcspring.2015.7146069).

[25] P. Wang *et al.*, "Location-partition-based Channel Allocation and Power Control Methods for C-V2X Communication Networks", *Wireless Networks*, vol. 26, no. 3, pp. 1563–1575, 2019 (https://doi.org/10.1007/s11276-019-02206-0).

[26] X. Xie, J. Shi, and Q. Yang, "Location Based Channel Resource Allocation for V2V Communications", *2022 IEEE 16th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, Xiamen, China, 2022 (https://doi.org/10.1109/asid56930.2022.9995793).

[27] METIS, "Mobile and Wireless Communications Enablers for the Twenty-twenty Information", ICT-317669-METIS/D6.1, 2013 [Online]. Available: https://cordis.europa.eu/docs/projects/cnect/9/317669/080/deliverables/001-METISD61v1pdf.pdf.

[28] K. Bąkowski, K. Wesołowski, and M. Rodziewicz, "Simulation Tools for the Evaluation of Radio Interface Technologies for IMT-Advanced and Beyond", in: *Simulation Technologies in Networking and Communications*, pp. 365–390, 2014 (https://doi.org/10.1201/b17650).

[29] V. Nurmela *et al.*, "METIS D1.2: Initial Channel Models Based on Measurements", ICT-317669-METIS/D1.2, 2014 [Online]. Available: https://www.researchgate.net/publication/262160344_METIS_D12_Initial_channel_models_based_on_measurements.

[30] ITU, "ITU-R Recommendation P.1411. Propagation Data and Prediction Methods for the Planning of Short-Range Outdoor Radiocommunication Systems and Radio Local Area Networks in the Frequency Range 300 MHz to 100 GHz", 2012.

Marcin Rodziewicz, Ph.D.
Institute of Radiocommunications
https://orcid.org/0000-0002-0487-1204
E-mail: marcin.rodziewicz@put.poznan.pl
Poznan University of Technology, Poznań, Poland
https://put.poznan.pl

Semantic Segmentation of Plant Structures with Deep Learning and Channel-wise Attention Mechanism

Mukund Kumar Surehli[1], Naveen Aggarwal[2], Garima Joshi[2], and Harsh Nayyar[2]

[1]*VIT-Bhopal University, Madhya Pradesh, India,*
[2]*Panjab University, Chandigarh, India*

https://doi.org/10.26636/jtit.2025.1.1853

Abstract — Semantic segmentation of plant images is crucial for various agricultural applications and creates the need to develop more demanding models that are capable of handling images in a diverse range of conditions. This paper introduces an extended DeepLabV3+ model with a channel-wise attention mechanism, designed to provide precise semantic segmentation while emphasizing crucial features. It leverages semantic information with global context and is capable of handling object scale variations within the image. The proposed approach aims to provide a well generalized model that may be adapted to various field conditions by training and tests performed on multiple datasets, including Eschikon wheat segmentation (EWS), humans in the loop (HIL), computer vision problems in plant phenotyping (CVPPP), and a custom "botanic mixed set" dataset. Incorporating an ensemble training paradigm, the proposed architecture achieved an intersection over union (IoU) score of 0.846, 0.665 and 0.975 on EWS, HIL plant segmentation, and CVPPP datasets, respectively. The trained model exhibited robustness to variations in lighting, backgrounds, and subject angles, showcasing its adaptability to real-world applications.

Keywords — channel-wise attention, computer vision, DeepLabV3+, deep learning, plant segmentation, semantic segmentation

1. Introduction

Computer vision-based segmentation methods have been used to address some data-rich agriculture problems. Segmentation of plant structures enables precise crop monitoring, disease detection, and weed management. It facilitates targeted interventions, optimizes usage of resources and contributes to sustainable production.

Semantic segmentation of plant structures involves classifying image pixels into distinct categories, such as leaves, stems, and fruits. Conventionally, thresholding for semantic segmentation of plant images involves differentiating between the foreground (green color) and background. Such an approach was employed in works [1] and [2]. This involved using such indices as excess green index (ExG) [3], normalized green-red difference index (NGRDI) [4], and color index of vegetation extraction (CIVE) [5] to enhance the green color of plants in the images.

The reliance of such methods on the green color limits their effectiveness and applicability to cases where the plant color differs significantly from the background. Additionally, variations in lighting conditions and reflections can impact segmentation accuracy, posing challenges for generalization across different scenarios [6]. Therefore, classic methods used for semantic segmentation of plant images pay attention to the features in an image and compare the differences between and/or gradients of pixels. These methods employ mathematical models and algorithms to identify regions of interest within an image. To identify these regions, common characteristics such as color, texture, and intensity are used.

These segmentation techniques, though simple, fast and memory-efficient, are more applicable to simple segmentation tasks. They require fine tuning for the specific use case and provide limited accuracy for complex scenes, which makes them considerably unsuitable for dealing with plant images.

On the other hand, deep learning-based methods perform segmentation-related tasks by employing neural networks to identify the vital features in an image [7]. These developments have resulted in some decent image segmentation models, boasting remarkable performance improvements over their classic predecessors.

2. Related Work

Research and development related to semantic segmentation focused on deep learning approaches and has resulted in the creation of various models relying on a wide array of architectures. The authors of [8] proposed a two-stage deep learning approach for plant disease detection. In the first stage, semantic segmentation models (U-Net [9], SegNet [10], and DeepLabV3+ [11]) were employed to extract plants from the input, with U-Net achieving the highest mean weighted intersection over union (mwIOU) of 0.9422. In the second stage, DeepLabV3+ outperformed the previous approach achieving mwIOU of 0.7379 for disease localization. Such an integrated model combining U-Net and DeepLabV3+ demonstrated robust performance. Unfortunately, the paper lacks a discussion on the generalization methods making it suitable to various crop species or diseases and omits

insights into computational costs – a crucial aspect for real-time segmentation tasks. These limitations should be taken into account while considering broader applicability of the proposed two-stage model.

Paper [12] introduces a semantic segmentation framework leveraging both real and synthetic data. The proposed approach employes a mask region-based convolutional neural network (R-CNN) model with a ResNet101 backbone [13] and a feature pyramid network (FPN) [14]. Synthetic images were generated from a dataset focusing on computer vision problems in plant phenotyping (CVPPP) encountered in the leaf segmentation challenge (LSC) [15]. Training the model on a dataset comprising both real and synthetic images, the authors achieved a leaf segmentation score of 90% on the A1 subset of the CVPPP dataset, with a mean score of 81% across the entire dataset.

Article [16] utilizes U-Net for semantic segmentation of leaf structures in plants. Leveraging the architecture's lightweight structure, as well as its computationally less intensive nature and fast inference, the authors trained it on the CVPPP-LSC dataset and achieved an intersection over union (IoU) of 90.56% and 98.69% on training and testing sets, respectively. Utilizing the U-Net architecture, the proposed model was resistant to varying input image dimensions. However, it is crucial to take note of the fact that the proposed approach is overly reliant on the training dataset.

The dataset contains images of *Arabidopsis thaliana* and *Nicotiana tabacum* (Tobacco) plants, with the structures being highly similar and all green. This affects the model's ability to generalize over a diverse range of plant structures and varying colors. Additionally, the images were captured in indoor conditions, in ideal lighting environments and have no structural overlap. This creates reservations concerning the segmentation quality of the model when used outdoors, where lighting, shadows, occlusions and position of the subject all could vary. In order to be used in real-world applications, the model would require rigorous tuning and diverse datasets.

The authors of [17] performed semantic segmentation on tall fig shrubs under real-world, open-field cultivation conditions. The proposed methodology made use of a convolutional neural networks (CNN) architecture inspired by SegNet with fewer trainable parameters. It was trained on a custom dataset comprising fig shrub images, captured from a drone, at a relatively high altitude and achieved an impressive accuracy of 93.84%. The introduced model was robust enough to handle varying outdoor visual conditions, such as shadows, occlusions, plant overlap, sunlight illumination. Additionally, owing to its smaller size, it was relatively computationally less intensive for inference purposes.

The literature review indicates that complexity of plant structures is an important factor, as plants contain structures of varying scale – from fine vein-like elements to the shape of the entire plant – making it increasingly difficult to have one model to detect them all. Illumination is another key factor, as variations in lighting conditions, especially in outdoor settings, affect the plants" appearance and visibility. The background is the next factor that needs to be taken into consideration, as images may often contain cluttered backdrops such as weeds, parts of other plants, soiling caused by wetting or drying, moss, etc. The availability of high-quality annotated data is another challenge. Annotation of plant images for segmentation purposes may be highly be labor-intensive and time-consuming Therefore, the process of creating large and diverse datasets poses a demanding challenge [18].

The culmination of these factors, from variations in outdoor conditions to the availability of data, creates a challenging scenario while developing the segmentation model. However, taking account of the dataset's inherent nature, it would be increasingly difficult to adapt the proposed model to segmenting plants that exhibit different structures, or to plant images that have been captured from a closer distance. The authors noted poor accuracy in scenes in which miscellaneous structures were visible along the fields or plants. Furthermore, the model's smaller size creates uncertainty regarding its ability to capture increasingly complex scenes for effective segmentation.

To overcome the "less-than-ideal" condition and over-reliance on training datasets, we introduce a DeepLabV3+ model coupled with a channel-wise attention mechanism referred to as "squeeze & excitation" (SE). The aforementioned model has been tested using our custom dataset, i.e. "botanic mixed set" [19], to evaluate the its generalization capabilities and applicability to data that is comprehensively unseen and different.

3. Methodology

The work presented in this paper introduces a robust semantic segmentation model with the ability to generalize over a wide array of plant species and handle various issues, as discussed in Section 1, by exploiting modern deep learning methods. The encoder's shallow layers represent the image as a low-level feature map presenting basic, simple, and local characteristics of an object in the image, such as edges, textures and corner points. The deeper layers output a high-level representation of the image, focusing on complex shapes and a deeper understanding of the global context. High-level features are often composites of multiple low-level features.

The DeepLabV3+ segmentation model was chosen based on its encoder-decoder architecture. The model makes use of the atrous spatial pyramid pooling (ASPP) module (fed with high-level features from the encoder), which internally makes use of atrous (or dilated) convolutions. Atrous convolutions differ from normal convolutions in the way that they introduce gaps in the kernel with a parameter called dilation rate. When these dilated kernels stride over image pixels, they may capture a wider field of view, thus producing a feature map that has a certain spatial context.

In the ASPP module, several atrous convolutions at different dilation rates are performed in parallel in order to obtain their corresponding feature maps. These maps, along with a global average pooling map, are concatenated to form the ASPP output. This output is rich in spatial context at different

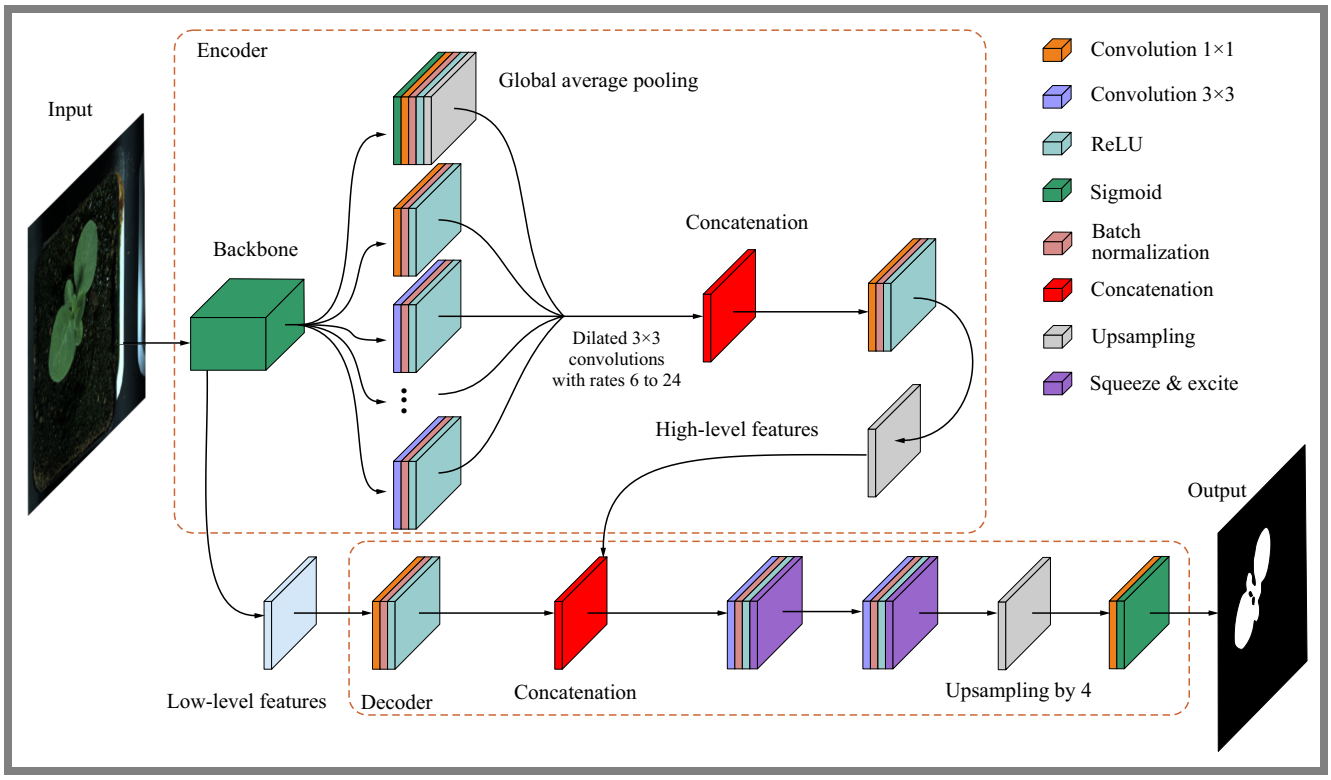

Fig. 1. Architecture of the implemented model.

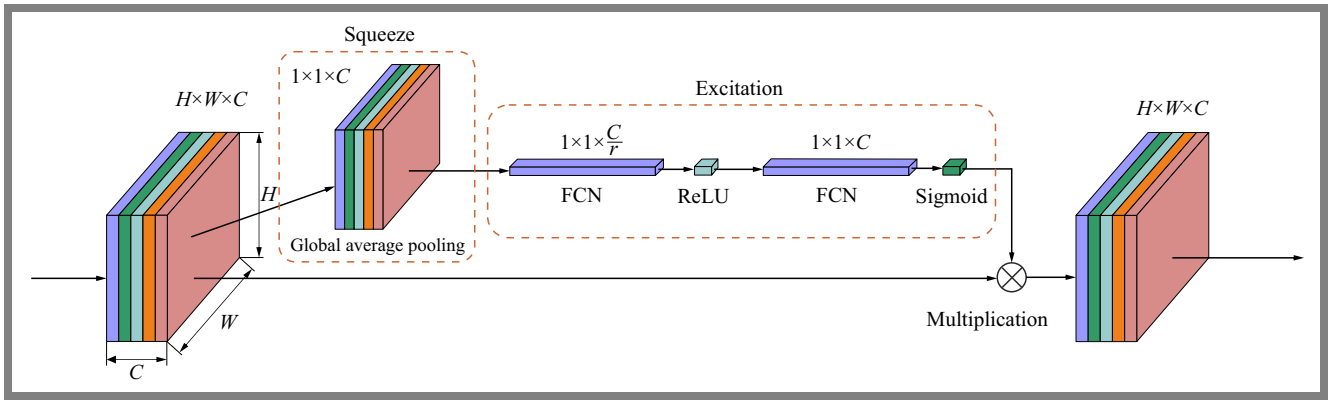

Fig. 2. Architecture of the squeeze and excitation (SE) module.

scales. High-level features from ASPP and low-level features from the encoder are concatenated together in the decoder to combine semantic information with increased spatial context.

Pixel-wise classifications are performed and the original input resolution is achieved with consequent 3×3 convolutions and bilinear up-sampling by 4. A simpler way to look at the architecture is to observe that at the model is initially aware of the position of a given object in the image (spatial information) but is not exactly aware of what that object is (lacking semantic information). As the input propagates forward, the model becomes aware of what the object is, thus gaining semantic information, but because of the repeated convolutions, it lacks global spatial information.

Segmentation models, with their goal being to assign class labels to every pixel in the image, require spatial information to delineate object boundaries and semantic information to differentiate object categories within the image. Thus, skip-connections are utilized to transport both spatial and semantic information (through ASPP) to the decoder. The decoder combines both and gets a concatenated feature map which not only knows what the object is, but also where exactly it is.

Figure 1 provides an overview of the model described in this paper. CNNs at each layer output feature maps that assist the network in extracting hierarchical information from the input. These feature maps are represented as tensors of $B \times C \times H \times W$ dimensionality, where B denotes the batch size, C denotes the number of channels, H denotes the height, and W denotes the width. Feature maps are representations of different features in an image, and directly affect the quality of output. The aim is to recalibrate these feature maps in such a way that, for a given task, they capture and highlight only those properties of an object that benefit the output.

A way of achieving feature recalibration is channel-wise attention. When CNN outputs feature maps, the channels within it are equally weighted. Since channels represent the derivation of different features from an input (convolutional filters), it is given that not all the channels within a feature map hold equal representational importance for a specific task. Paper [20] proposed a method of applying channel-wise weights to describe their representativeness. Thus, more important features are amplified the less useful ones are suppressed. This method involved using a SE module responsible for the squeeze and excite operation.

Figure 2 illustrates the SE operation that is employed in this study. For the squeeze operation, consider the feature map as a tensor of dimensionality $C \times H \times W$. A channel descriptor is formed here by aggregating the input feature map across its spatial dimensions $H \times W$, forming a single numerical value. The goal of this aggregation is to capture global information about the feature map in a channel-wise manner. Global average pooling is a way to perform these aggregations. The output of the squeeze operation is in the $C \times 1 \times 1$ dimensions and is forwarded to the excite operation.

The excite operation consists of two fully convolutional network (FCN) blocks with a ReLU and sigmoid activation, respectively. The first block performs dimensionality reduction on C channels by a reduction factor r, with the goal of this operation being to decrease computational complexity and maintain global information at smaller scales. The input is now reduced to dimensions of $\frac{C}{r} \times 1 \times 1$ and is passed onto the next FCN block for scaling the reduced map to the original dimension of $C \times 1 \times 1$. The output of the excitation operation is a set of scaling factors for each channel, represented by a weighted tensor of dimensions $C \times 1 \times 1$.

The weighted tensor can now be multiplied with the original feature map with dimensions $C \times H \times W$ to obtain the output of the SE module, i.e. a re-calibrated feature map. Within this study, SE modules with a reduction factor r of 8 have been incorporated in the final 3×3 convolutions, following the fusion of spatial and semantic information.

3.1. Implementation Details

The segmentation model was implemented with TensorFlow and Keras libraries. It was trained using the Nvidia Tesla T4 unit with 16 GB GPU memory. The model follows a DeepLabV3+ architecture with a choice of backbone between ResNet50 [13] and Xception [21], pre-trained on ImageNet weights [22]. For training datasets data augmentations applied as detailed in Section 4.

The training paradigm consisted of the Adam optimizer with a learning rate ranging from 1E–4 to 1E–7, batch-size of 16 and 80 epochs. The process was further configured with ReduceLROnPlateau with 5 epochs, a factor of 0.1 and early stopping with a patience of 20 epochs. L2 regularization was also used in the SE module with a factor of 1E–4. Incorporating an ensemble approach, an additional training methodology leveraged progressive refinement of weights, as depicted in Fig. 3.

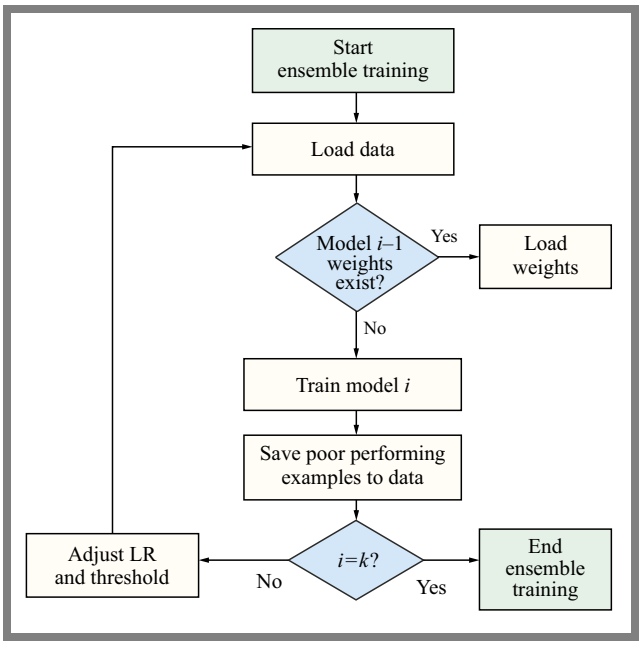

Fig. 3. Flowchart depicting ensemble training with i as the current model index and k representing the total number of models.

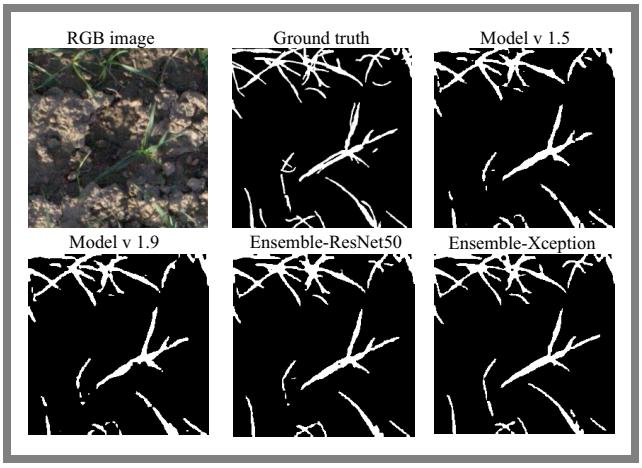

Fig. 4. Evaluation on EWS dataset.

Each subsequent model in the ensemble was initialized with the weights of the previous model. To increase robustness, poor performing examples present in the validation set, with the intersection over union (IoU) value below a certain threshold, were identified and oversampled at the end of each model's training. This intensified the model's subsequent exposure to challenging instances. To expedite model convergence, the learning rate for each new model in the ensemble was decreased (see Eq. (1)), aiding in achieving a dynamic adaptation mechanism. Additionally, the initial IoU threshold was increased, promoting a more deliberate learning process using the following equation:

$$\alpha_i = \frac{\alpha}{2^{(i-1)}}\,, \tag{1}$$

where: α_i – learning rate for model I, α – initial learning rate for the ensemble, and i – index of current model in the ensemble.

Tab. 1. Dataset details.

Dataset	Origin	No. of images	Plants	Imaging equipment	Conditions
EWS [23]	Eschikon, CH	190	Wheat	Canon 5D Mark II	Outdoor
CVPPP [15]	Various	810	Arabidopsis, tobacco	Varied cameras	Indoor
HIL [24]	Aarhus University, DK	144	Plant seedlings	Not specified	Indoor
BMS [19]	Chandigarh, IN	47	Various	Nikon D3300	Outdoor

4. Results

For training and evaluation of the model, three publicly available datasets were used: Eschikon wheat segmentation (EWS) [23], humans in the loop (HIL) plant segmentation [24] and computer vision problems in plant phenotyping (CVPPP) [15]. For a final evaluation with less than-ideal real-world conditions, a custom dataset called botanic mixed set (BMS) [19] was developed. The evaluation performed on the custom dataset aims to test the reliance of the model on the training datasets used, and to observe the model's ability to adapt to changing illumination, angle, and distance to subject within the images.

The datasets detailed in Tab. 1 are annotated with binary masks for soil/background and plant regions. The CVPPP dataset features subsets A1 to A4, offering distinct experimental settings for *Arabidopsis thaliana* and *Nicotiana tabacum* (Tobacco). Additionally, it undergoes a custom subset, combining subsets A1, A2 and A4 (totaling 267 images). The evaluation set for CVPPP consists of 63 images representing all of the subsets.

Data augmentation is a pivotal technique in enhancing the training efficiency of deep neural networks by artificially increasing the size of training set through transformations applied to the existing dataset. In study [25], various data augmentation techniques were tested within the context of CNNs, revealing their substantial impact on model training and evaluation. The chosen augmentation strategy plays a crucial role in improving the model's robustness by exposing it to a diverse range of scenarios. For this paper, data augmentation served the dual purpose of addressing the limited size of sourced datasets and aiding the model in effectively generalizing in response to previously unseen data.

Implemented through the augmentations Python library, the data augmentation process involved applying six transformations to each example in the dataset, resulting in six modified copies alongside the original example. This approach effectively increased the dataset size by a factor of six. The transformations included horizontal and vertical flipping, channel shuffling, random adjustments to brightness, contrast, rotation of the image by 45°, and a random crop. The resultant dataset was resized to 256 by 256 pixels.

It is crucial to note that the input images and masks underwent normalization to the [0, 1] range. Furthermore, the images were standardized using a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225] to adhere to ImageNet specifications. This pre-processing ensured compatibility and consistency in model training.

4.1. Metrics

For assessing performance of the model during the training and evaluation phases, intersection over union (IoU) and Dice coefficient were used. IoU, also known as the Jaccard index, is the main performance tool used in this work, and is a popular metric for segmentation tasks. It measures the accuracy of localizing objects by calculating the intersection of the predicted mask and the ground truth mask, dividing that by the union of the two:

$$IoU(A, B) = \frac{A \cap B + x}{A \cup B + x} . \tag{2}$$

Dice coefficient, also known as the Dice similarity index or Dice score, is another metric for image segmentation. It is used to quantify the similarity or overlap between two sets. In the context of image segmentation, it can be used to compare pixel-wise agreement between the predicted mask and its corresponding ground truth:

$$Dice\,(A, B) = \frac{2 \times (A \cap B) + x}{(A \cup B) + x} . \tag{3}$$

Dice loss is the loss function of the proposed model, derived from the Dice coefficient. This loss function encourages the model to produce masks that have a higher area of overlap with ground truth masks:

$$DiceLoss\,(A, B) = 1 - \frac{2 \times (A \cap B) + x}{(A \cup B) + x} . \tag{4}$$

In Eqs. (2) – (4), A is the predicted mask, B stands for the ground truth mask, and x is the smoothing factor.

The focus of the experiments was to enhance the generalization ability of the model across varying conditions. Dice loss, IoU, mean IoU (mIoU) and mean Dice coefficient (mDice) are the evaluation metrics used. It should be noted that the benchmarks presented further are results of training with a ResNet50 and Xception backbone, pretrained with the ImageNet weights.

4.2. Evaluation on EWS Dataset

In the process of hyperparameter and dataset tuning, an array of model versions was developed to obtain the best results. The best validation results achieved, measured based on IoU,

Tab. 2. Performance evaluation on EWS dataset.

Model version	Training	Backbone	IoU	Dice loss	mIoU
1.0	Singular	ResNet50	0.705	0.176	0.578
1.1	Singular	ResNet50	0.741	0.151	0.607
1.2	Singular	ResNet50	0.753	0.143	0.621
1.3	Singular	ResNet50	0.755	0.141	0.628
1.4	Singular	ResNet50	0.763	0.134	0.621
1.5	Singular	ResNet50	0.768	0.131	0.629
1.6	Singular	ResNet50	0.766	0.132	0.625
1.7	Singular	ResNet50	0.762	0.152	0.625
1.7.1	Singular	ResNet50	0.413	0.486	0.643
1.8	Singular	ResNet50	0.758	0.170	0.621
1.9	Singular	ResNet50	0.767	0.143	0.626
-	Ensemble	ResNet50	0.846	0.084	0.828
-	Ensemble	Xception	0.842	0.087	0.826

Tab. 3. Benchmark results and comparison with other papers related to EWS dataset.

Benchmark	IoU
Rico-Fernández *et al.* [26]	0.691
Zenkl *et al.* [23]	0.775
Yu *et al.* [27]	0.666
Sadeghi-Tehran *et al.* [28]	0.638
Proposed model	0.846

and their comparison with other publications, are presented in Tab. 3.

In [26], spatial context is provided in the form of a 5×5 window around individual pixels translated into CIELUV color space and input into a support vector classifier (SVC). Paper [23] used a DeepLabV3+ model with a ResNet50 backbone, feeding extra features as supplementary inputs. SVC was employed in the decision tree with preliminary weather state classification in [27]. The authors of [28] used a random forest classifier with the input having the form of 21 different color features. Note that the methods from [26]–[28] were reverse-engineered and tested by authors of [23] to obtain benchmarks on the EWS dataset. Table 3 presents a comparison of metrics between the aforementioned publications, while Tab. 2 presents a comparison of results between different model variations proposed in this paper.

As one may notice from Tab. 2, there is a significant decrease in the metrics in model v1.7.1. This decrease may be attributed to the number of dilated convolutional layers in the ASPP module. Model v1.7.1 saw an increase to 4 dilated convolution layers, but their respective rates were reduced to 4, 8, 12 and 16, in contrast to the higher dilation rates used in other model variations. This highlights the observation that the choice of dilation rates and the number of dilated convolutions may exert a significant impact on model performance and segmentation quality. The contrast between the Dice loss metric in model v1.5 and v1.9 is visible in that the former was configured with 3 dilated convolutions with rates of 12, 24, 36, and the latter was configured with 5 dilated convolutions with rates of 6, 12, 18, and 24. While IoU between the two remained close (0.768 vs. 0.767), the Dice loss varied (0.131 vs. 0.143). It should be noted that every other adjustable hyperparameter was kept identical for both model variations.

Since ASPP is crucial for capturing spatial information, lower dilation rates may help in paying attention to intricate details in the scene, while higher rates can assist in capturing wider plant variations. Despite the higher loss in model v1.9, it is able to capture the intricate details better, but may require more computation on account of more parallel dilated convolutions.

The ensemble training approach making use of ResNet50 yields superior metrics, as the progressive refinement of weights is coupled with increased exposure to sub-optimal instances. It leads to a final model characterized by superior segmentation quality and heightened resilience to diverse conditions. The Xception-backed ensemble model achieves slightly lower metrics than ResNet50, but was increasingly efficient at capturing the finer-grained details in the scene, paving the way for better segmentation quality.

4.3. Evaluation on HIL Plant Segmentation Dataset

The training process for HIL plant segmentation dataset (HIL-PS) followed the same procedure as EWS. The results of different model variations are presented in Tab. 4. While the model has demonstrated satisfactory performance on other datasets, it is crucial to take note of dataset-specific attributes. The lower performance metrics on the HIL-PS dataset can be attributed to a distinctive characteristic, specifically the relatively smaller size of plant specimens. Despite the model's capability to capture fine-grained details, the diminutive

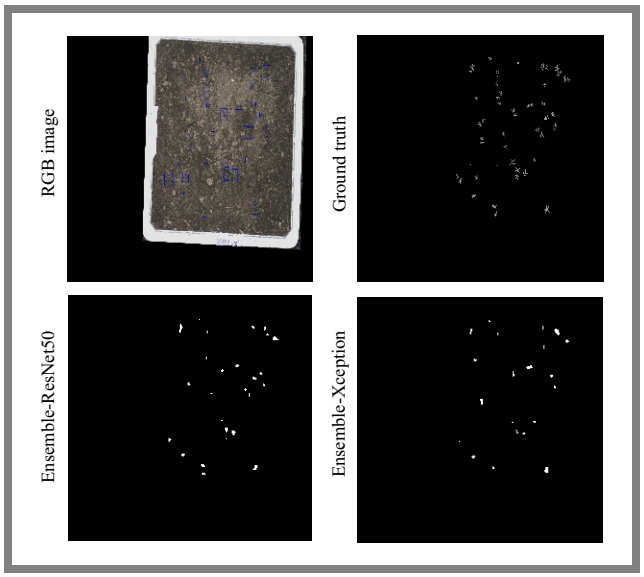

Fig. 5. Evaluation on HIL – diminutive samples.

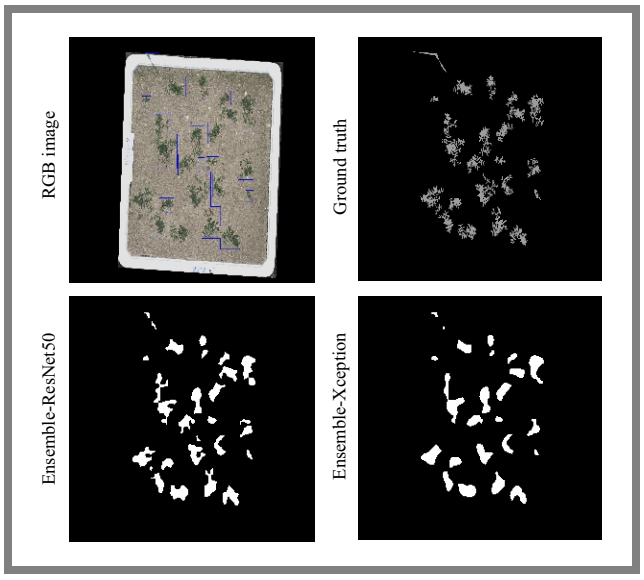

Fig. 6. Evaluation on HIL – optimal size samples.

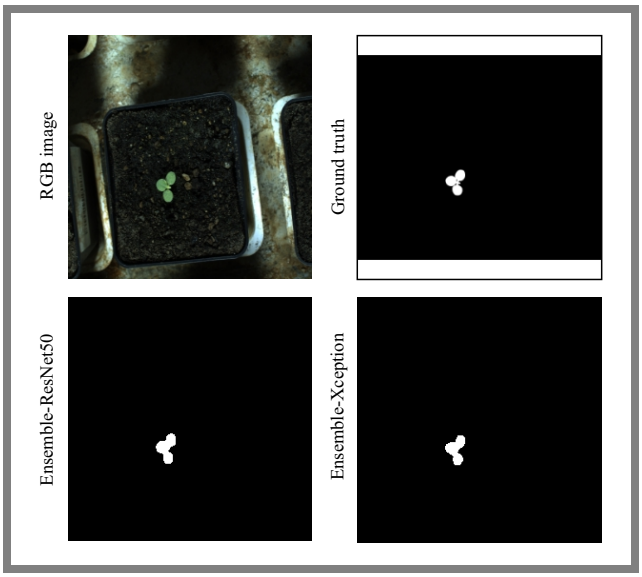

Fig. 7. Evaluation on CVPPP.

Fig. 8. Evaluation on CVPPP dataset.

proportions of the plants result in scenes with limited informative content. The model excels when dealing with scenes abundant with acquirable information, but faces difficulties in preserving segmentation quality for smaller-sized samples. The approach to ensemble training proved to be a pivotal enhancement. It not only led to improvements in performance metrics but also showcased superior segmentation quality for diminutive and optimally sized samples. The segmentation outcomes from the optimal model (ensemble) for this specific dataset are depicted in Figs. 5–6.

4.4. Evaluation on CVPPP Dataset

This subsection presents the results of models trained on different splits (A1, A2, A3, A4, custom split) of the CVPPP dataset. The evaluation split is formulated from a combination of splits known as A1 to A4 (63 images in total) – see Tab. 5. In Tab. 6, the precise dataset splits used for model training determine the observed variance in outcomes. The A1 split was mostly made up of *Arabidopsis thaliana* images, with plant specimens grown in pots, frequently accompanied by a dirt surface/background covered with green colored moss. This unique environment complexity most likely contributed to the poorer IoU and Dice loss metrics. The A3 split, on the other hand, revealed a data restriction with just 27 images of *Nicotiana tabacum* plants. Furthermore, the photos included plants with lower proportions and complex backgrounds, which made proper model training difficult.

Despite these obstacles, the A3-trained model demonstrated exceptional generalization skills on the CVPPP evaluation set, showcasing its adaptability to conditions exceeding the training limits. In contrast, the model trained on the custom split displayed exceptional metrics – as listed in Tab. 6 – for splits A1 to A4. Suitably sized and augmented splits increased its diversity and facilitated effective model generalization. Using an ensemble paradigm for custom split training has led to a small improvement in metrics and better generalization

Tab. 4. Performance evaluation on HIL-PS dataset.

Model version	Training	Backbone	IoU	Dice loss	mIoU
1.0	Singular	ResNet50	0.288	0.567	0.372
1.1	Singular	ResNet50	0.516	0.356	0.420
1.2	Singular	ResNet50	0.550	0.307	0.472
1.3	Singular	ResNet50	0.550	0.306	0.469
1.4	Singular	ResNet50	0.547	0.309	0.458
–	Ensemble	ResNet50	0.665	0.202	0.547
–	Ensemble	Xception	0.646	0.217	0.494

Tab. 5. Performance evaluation on CVPPP dataset.

Dataset	Training	Backbone	IoU	Dice loss	mIoU
A1	Singular	ResNet50	0.454	0.387	0.215
A2	Singular	ResNet50	0.915	0.044	0.635
A3	Singular	ResNet50	0.451	0.362	0.652
A4	Singular	ResNet50	0.921	0.043	0.812
Custom split	Singular	ResNet50	0.957	0.051	0.853
Custom split	Ensemble	ResNet50	0.975	0.013	0.859
Custom split	Ensemble	Xception	0.972	0.015	0.856

Tab. 6. Split-wise evaluation results.

Dataset	Training	Backbone	Evaluation – split	mIoU	mDice
Custom split	Singular	ResNet50	A1	0.930	0.964
			A2	0.816	0.884
			A3	0.880	0.919
			A4	0.915	0.955
Custom split	Ensemble	ResNet50	A1	0.937	0.967
			A2	0.844	0.907
			A3	0.889	0.924
			A4	0.921	0.958
Custom split	Ensemble	Xception	A1	0.922	0.960
			A2	0.787	0.861
			A3	0.898	0.943
			A4	0.911	0.953

across all splits. This shows the impact of dataset variables on model's performance and illustrates the efficiency of strategic augmentation and ensemble techniques in generalization across plant diversity. Figures 7-8 show segmentation results on the optimal model for this dataset.

4.5. Evaluation on BMS Dataset

The best-performing models from singular and ensemble training on EWS, HIL, and CVPPP datasets were further tested on the botanic mixed set (BMS). This was done to assess how well they could adapt to different training sets. Mean intersection over union (mIoU) and mean Dice coefficient (mDice) were the metrics used for this evaluation. The goal was to understand if these models could perform consistently across a diverse range of botanical specimens. The results shed light on the adaptability and reliability of the models in diverse settings. Through the metrics presented in Tab. 7, it is evident the EWS-trained model is the best performer. The model was able to capture a variety of plant structures in the image, such as leaves, long stems, etc. but struggled with capturing diminutive structures and often under- or over-classified the objects. This could be attributed to the nature of the data the model was trained on. The HIL-trained

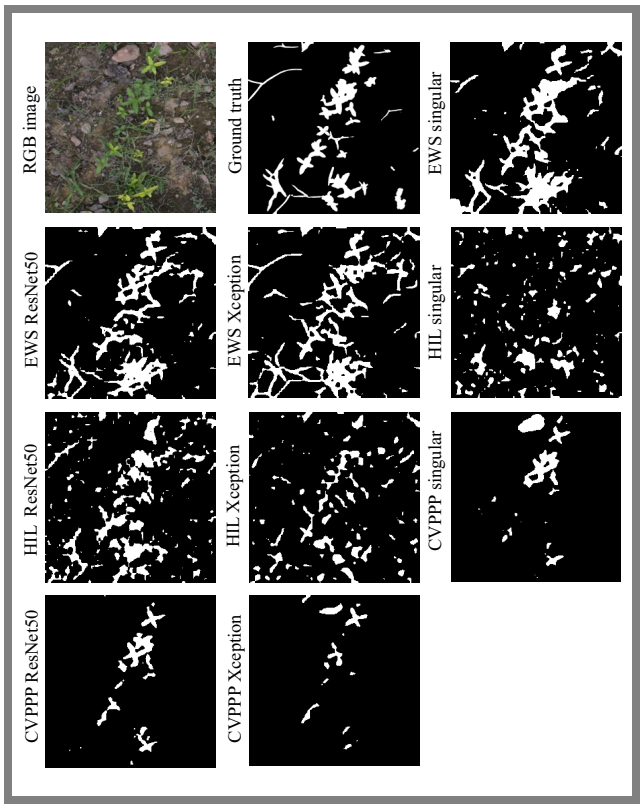

Fig. 9. Evaluation on BMS dataset.

Tab. 7. Performance evaluation on BMS dataset.

Dataset	Training	Backbone	mIoU	mDice
EWS	Singular	ResNet50	0.504	0.642
	Ensemble	ResNet50	0.503	0.643
	Ensemble	Xception	0.508	0.648
HIL	Singular	ResNet50	0.243	0.369
	Ensemble	ResNet50	0.385	0.524
	Ensemble	Xception	0.278	0.423
CVPPP	Singular	ResNet50	0.287	0.394
	Ensemble	ResNet50	0.248	0.332
	Ensemble	Xception	0.184	0.259

model was comparatively better at capturing the diminutive species but struggled with global context, hence could not segment larger plant structures. It was also observed that the model significantly struggled with complex backgrounds in the input images. The CVPPP-trained model was remarkable at capturing leaf structures and finer details but struggled with plants that did not resemble the structure of plants in its training set, i.e. long stems. It was observed that the ensemble models, even when not trained with augmented data, demonstrated superior metrics and segmentation quality in certain scenarios. This underscores their exceptional capacity to generalize across a dataset despite having fewer examples – see Fig. 9.

5. Limitations and Future Work

Deep learning models are capable of fitting large amounts of training data for generalization purposes. Their learning capabilities make them extremely susceptible to overfitting or underfitting in a scenario in which the dataset is smaller in size. The goal is to find a balance between the two and generalize well for a salient task. DeepLabV3+, while being a powerful semantic segmentation model, is relatively large and complex. Training with larger datasets and batch sizes may require significant amounts of computational power. The architecture also requires large datasets for effective training, which can be a bottleneck as datasets with good ecological diversity and quality annotations are difficult to find in the public domain. Adapting the architecture to a particular domain requires significant data and hyper-parameter tuning, both of which are time consuming.

The real-world generalization ability of each specific dataset-trained model was consistently meeting the expected standards. Factors impacting a given scene, such as lightning, background, angle of view, only minimally impacted the model's ability to recognize objects of interest. While each model had the ability to perform segmentation despite scene-related conditions, it is essential to acknowledge that not all models provided decent generalization for real-world scenarios. The training datasets lacked the size and diversity for such a precise task. Addressing such a limitation would require obtaining a dataset with more ecological diversity.

Plant specimens of smaller sizes also posed a limitation for the proposed method. Despite selected approaches to data augmentation and model-tuning, the segmentation of smaller plant structures proved to be challenging. Further research is required to address this constraint and potential directions may include the exploration of scale-aware models. A combined dataset incorporating examples from EWS, HIL, CVPPP and BMS could be leveraged for model training. Based on the results, relevant examples to maintain botanic diversity could be included or excluded from this new dataset to discourage class imbalance.

6. Conclusions

The proposed architecture performed semantic segmentation of plant images by incorporating an extended DeepLabV3+ model with a channel-wise attention mechanism. The work aimed to address generalization- and scene-related variations affecting task of segmenting plant images. The proposed model offers a powerful semantic segmentation solution emphasizing the features and leveraging semantic information with global context.

Several datasets were used to train their own versions of the model, which were further tested on the custom BMS dataset used for evaluation purposes. Adding an attention mechanism offered an increase in the quality of segmentation, when compared to earlier model-variations during dataset

and hyper parameter tuning. The models exhibited robustness to variations in lighting, backgrounds, and subject angles, showcasing their adaptability to real-world applications.

Additionally, even better results could be obtained by the inclusion of ensemble training. Through the application of ensemble approaches, the models showed exceptional resilience to real-world differences in lighting, backdrops, and subject angles, as well as outstanding generalization skills. The segmentation performance was much improved by relying on the ensemble training approaches.

Further research directions will be focused on improvements to the model's architecture and dataset composition. The current efforts are intended to expand on the accomplishments of the past and improve the model's ability to handle various problems that arise in practical implementations.

References

[1] M.A. Castillo-Martinez *et al.*, "Color Index Based Thresholding Method for Background and Foreground Segmentation of Plant Images", *Computers and Electronics in Agriculture*, vol. 178, 2020 (https://doi.org/10.1016/j.compag.2020.105783).

[2] D. Riehle, D. Reiser, and H.W. Griepentrog, "Robust Index-based Semantic Plant/background Segmentation for RGB-images", *Computers and Electronics in Agriculture*, vol. 169, 2020 (https://doi.org/10.1016/j.compag.2019.105201).

[3] D.M. Woebbecke, G.E. Meyer, K.V. Bargen, and D.A. Mortensen, "Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions", *Transactions of the ASAE*, vol. 38, pp. 259–269, 1995 (https://doi.org/10.13031/2013.27838).

[4] E.R. Hunt *et al.*, "Evaluation of Digital Photography from Model Aircraft for Remote Sensing of Crop Biomass and Nitrogen Status", *Precision Agriculture*, vol. 6, pp. 359–378, 2005 (https://doi.org/10.1007/s11119-005-2324-5).

[5] D. Zhang *et al.*, "A Universal Estimation Model of Fractional Vegetation Cover for Different Crops Based on Time Series Digital Photographs", *Computers and Electronics in Agriculture*, vol. 151, pp. 93–103, 2018 (https://doi.org/10.1016/j.compag.2018.05.030).

[6] J. Singh and H. Kaur, "Plant Disease Detection Based on Region-based Segmentation and KNN Classifier", *Proc. of the International Conference on ISMAC in Computational Vision and Bio-Engineering*, pp. 1667–1675, 2018 (https://doi.org/10.1007/978-3-030-00665-5_154).

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 640–651, 2017 (https://doi.org/10.1109/TPAMI.2016.2572683).

[8] L.G. Divyanth, A. Ahmad, and D. Saraswat, "A Two-stage Deep-learning Based Segmentation Model for Crop Disease Quantification Based on Corn Field Imagery", *Smart Agricultural Technology*, vol. 3, 2023 (https://doi.org/10.1016/j.atech.2022.100108).

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Lecture Notes in Computer Science*, vol. 9351, pp. 234–241, 2015 (https://doi.org/10.1007/978-3-319-24574-4_28).

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-decoder Architecture for Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2017 (https://doi.org/10.1109/TPAMI.2016.2644615).

[11] L.-C. Chen *et al.*, "Encoder-decoder with Atrous Separable Convolution for Semantic Image Segmentation", *Computer Vision – ECCV*, vol. 11211, pp. 833–851, 2018 (https://doi.org/10.1007/978-3-030-01234-2_49).

[12] D. Ward, P. Moghadam, and N. Hudson, "Deep Leaf Segmentation Using Synthetic Data", *ArXiv*, 2018 (https://doi.org/10.48550/arXiv.1807.10931).

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2016 (https://doi.org/10.1109/CVPR.2016.90).

[14] T.-Y. Lin *et al.*, "Feature Pyramid Networks for Object Detection", *ArXiv*, 2017 (https://doi.org/10.48550/arXiv.1612.03144).

[15] M. Minervini, A. Fischbach, H. Scharr, and S.A. Tsaftaris, "Finely-grained Annotated Datasets for Image-based Plant Phenotyping", *Pattern Recognition Letters*, vol. 81, pp. 80–89, 2016 (https://doi.org/10.1016/j.patrec.2015.10.013).

[16] M. Trivedi and A. Gupta, "Automatic Monitoring of the Growth of Plants Using Deep Learning-based Leaf Segmentation", *International Journal of Applied Science and Engineering*, vol. 18, 2021 (https://doi.org/10.6703/IJASE.202106_18(2).003).

[17] J. Fuentes-Pacheco *et al.*, "Fig Plant Segmentation from Aerial Images Using a Deep Convolutional Encoder-decoder Network", *Remote Sensing*, vol. 11, 2019 (https://doi.org/10.3390/rs11101157).

[18] S. Sharma, K. Verma, and P. Hardaha, "Implementation of Artificial Intelligence in Agriculture", *Journal of Computational and Cognitive Engineering*, vol. 2, pp. 155–162, 2023 (https://doi.org/10.47852/bonviewJCCE2202174).

[19] M.K. Surehli, N. Aggarwal, and G. Joshi, "Botanic Mixed Set", *GitHub*, 2023 (https://github.com/mukund-ks/botanic-mixed-set.git).

[20] J. Hu, L. Shen, G. Sun, "Squeeze-and-Excitation Networks", *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018 (https://doi.org/10.1109/CVPR.2018.00745).

[21] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258, 2017 (https://doi.org/10.1109/CVPR.2017.195).

[22] J. Deng *et al.*, "ImageNet: A Large-scale Hierarchical Image Database", *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, 2009 (https://doi.org/10.1109/CVPR.2009.5206848).

[23] R. Zenkl *et al.*, "Outdoor Plant Segmentation with Deep Learning for High-throughput Field Phenotyping on a Diverse Wheat Dataset", *Frontiers in Plant Science*, vol. 12, 2022 (https://doi.org/10.3389/fpls.2021.774068).

[24] Humans in the Loop, "Plant Segmentation Dataset", (https://humansintheloop.org/resources/datasets/plant-segmentation).

[25] E. Pereira, G. Carneiro, and F.R. Cordeiro, "A Study on the Impact of Data Augmentation for Training Convolutional Neural Networks in the Presence of Noisy Labels", *35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Natal, Brazil, 2022 (https://doi.org/10.1109/SIBGRAPI55357.2022.9991791).

[26] M. Rico-Fernandez *et al.*, "A Contextualized Approach for Segmentation of Foliage in Different Crop Species", *Computers and Electronics in Agriculture*, vol. 156, pp. 378–386, 2019 (https://doi.org/10.1016/j.compag.2018.11.033).

[27] K. Yu *et al.*, "An Image Analysis Pipeline for Automated Classification of Imaging Light Conditions and for Quantification of Wheat Canopy Cover Time Series in Field Phenotyping", *Plant Methods*, vol. 13, 2017 (https://doi.org/10.1186/s13007-017-0168-4).

[28] E. David *et al.*, "Global Wheat Head Detection (GWHD) Dataset: A Large and Diverse Dataset of High-resolution RGB-labelled Images to Develop and Benchmark Wheat Head Detection Methods", *Plant Phenomics*, 2020 (https://doi.org/10.34133/2020/3521852).

Mukund Kumar Surehli, B.Tech.
School of Computing Science and Engineering
E-mail: mukund.28.k@gmail.com
VIT-Bhopal University, Madhya Pradesh, India
https://vitbhopal.ac.in

Naveen Aggarwal, Ph.D.
University Institute of Engineering and Technology
E-mail: navagg@pu.ac.in
Panjab University, Chandigarh, India
https://puchd.ac.in

Garima Joshi, Ph.D.
University Institute of Engineering and Technology
E-mail: joshigarima5@yahoo.com
Panjab University, Chandigarh, India
https://puchd.ac.in

Harsh Nayyar, Ph.D.
Department of Botany
E-mail: nayarbot@pu.ac.in
Panjab University, Chandigarh, India
https://puchd.ac.in

Ultra-wideband Antenna System Design for Future mmWave Applications

Muhannad Y. Muhsin[1], Zainab S. Muqdad[2], Asaad H. Sahar[3], Zainab F. Mohammad[1], and Hussam AL-Saedi[1]

[1] *University of Technology – Iraq, Baghdad, Iraq,*
[2] *Mustansiriyah University, Baghdad, Iraq,*
[3] *University of Baghdad, Baghdad, Iraq*

https://doi.org/10.26636/jtit.2025.1.1951

Abstract — An ultra-wideband planar four-element multiple-input multiple-output (MIMO) antenna array for millimeter wave (mmWave) 5G applications is presented in this article, characterized by a simple structure and diverse performance capabilities. The antenna system operates in the 20 GHz band (ranging from 42.3 to 63.3 GHz), with a high gain of 7.8 dB. The compact size of 25 × 25 mm makes it suitable for being integrated with various telecommunication devices used in a number of mmWave applications. The antenna's elements are placed orthogonally, achieving great isolation of over 24 dB. The performance of the proposed antenna was analyzed in terms of its s parameters, gain, efficiency, radiation patterns, and MIMO diversity characteristics, including the envelope correlation coefficient (ECC), diversity gain (DG), and mean effective gain (MEG).

Keywords — 5G, antenna array, mmWave, UWB

1. Introduction

Fifth-generation communication networks represent a substantial advancement in wireless technologies, providing considerable potential for high-speed data transmission and ultralow latency [1]–[3]. The mmWave spectrum provides extensive capacity, allowing data rates to surpass those of former cellular networks [4]. The MIMO communication technology is used at high frequencies to achieve such a high data rate. In light of the above, it is necessary to design a system of MIMO antennas that relies on multiplexing technologies and spatial diversity to improve reliability, data throughput and coverage [5], [6].

Researchers aim to improve various antenna performance parameters, including bandwidth, compactness, efficiency, gain, and diversity characteristics. Many techniques are deployed to optimize performance, including substrate selection, dielectric lens, multi-element configurations, corrugation, and mutual coupling reduction [7], [8].

The substrate utilized in the design of each antenna is of key significance. A substrate with reduced relative permittivity and lower loss tangent will enhance gain while minimizing power losses [9]. A corrugated construction, which eliminates the metallic part of the edge radiator, is capable of improving the front-to-back ratio and the antenna's bandwidth [10]. Furthermore, the multi-element technique improves gain, efficiency, and bandwidth. Consequently, structures relying on the abovementioned solutions are capable of achieving elevated gain and bandwidth levels – a feature that one antenna only cannot offer. Moreover, a dielectric lens is capable of transmitting electrostatic radiation in a non-directional manner, thereby enhancing the antenna's directivity and gain, as shown in [11], [12].

Techniques relied upon to reduce mutual coupling decrease the impact that numerous elements exert on the system's performance. This method, referred to as an isolation technique, plays and important role in optimizing the performance of good diversity in MIMO structures [13], [14]. In addition, high gain is expected in mmWave antenna designs due to the increased free space path loss observed at these frequencies. Therefore, achieving decent performance of ultra-wideband MIMO antenna systems (in terms of high isolation and reasonable radiation characteristics) while maintaining their compact size and simple structure remains a challenge.

2. Related Works

In [15], a four-element MIMO array has been developed, where the antenna measuring only 30 × 35 × 0.76 mm. It operates at 25.5 – 29.6 GHz, with a gain of 8.3 dBi. The antenna design from [16] does not include decoupling surfaces and has a patch array. It offers a gain of 13.1 dBi within a frequency range of 26.94 – 31.08 GHz, with isolation below 14 dB.

In [17], the authors employed the DGS technique to provide robust isolation between an input port of the MIMO antenna array. In [18], a radiating element based on multiple arcs has been developed for mmWave MIMO applications, achieving a bandwidth of 23.5 – 38 GHz and offering a high isolation level of above 23 dB by arranging the radiating parts in an orthogonal configuration. Nevertheless, the overall profile of the antenna is rather large.

In [19], the researchers designed a tree-shaped antenna with multiple arcs to provide a wide broadband response of 23 – 40 GHz, an overall gain of 8 dBi and good isolation.

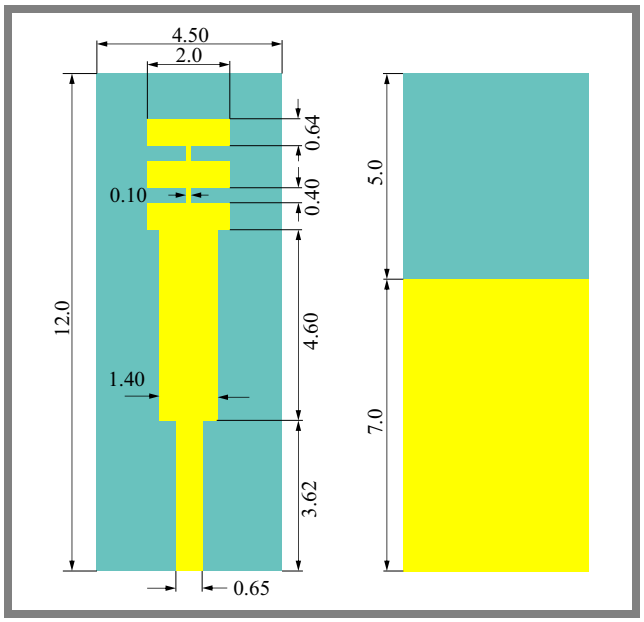

Fig. 1. Single antenna element: a) top view and b) back view.

A 4-element MIMO system designed for the 5G frequency spectrum of 27.5 – 40 GHz, with an entire dimension of 158 × 77.8 mm, was presented in [20], while paper [21] proposed an 8-port MIMO antenna array for the 27.5 – 29.5 GHz frequency range. In articles [22]–[25], metamaterial is used to achieve enhanced port isolation. A MIMO antenna array supporting 5G is presented in [26], achieving an array gain of 5 dB with a frequency band better than 26 – 39 GHz. In papers [27] and [28], another MIMO antenna array is proposed with high bandwidth and good radiation characteristics.

This article presents an ultra-wideband four-element MIMO array for mmWave applications featuring a simple structure and compact size of 25 × 25 mm. High isolation exceeding 24 dB is achieved thanks to orthogonal polarization diversity and excellent MIMO performance. It is a four-antenna MIMO system operating in the range of 42.3 to 63.3 GHz (20 GHz bandwidth). The antenna element is made up of two microstrip line steps (with different widths and lengths) and three symmetrical ladder steps located at the top. The proposed structure may be easily integrated with various devices used in mmWave applications. The CST Microwave Studio was used in the simulations.

3. Design

For micro-wave applications, printed antennas are optimal solutions due to their low profile, low cost, compact dimensions, as well as good balance between performance and manufacturing complexity. The proposed antenna utilizes the RT-5880 substrate, characterized by a loss tangent of 0.0009 and a relative permittivity of 2.2, with a thickness of 0.8 mm.

Figure 1 illustrates the shape and various parameters of the design. The size of the single antenna unit is 4.5 × 12 mm. The top view is presented in Fig. 1a, showing two microstrip line steps (with different widths and lengths) and three symmetrical ladder steps at the top of the antenna.

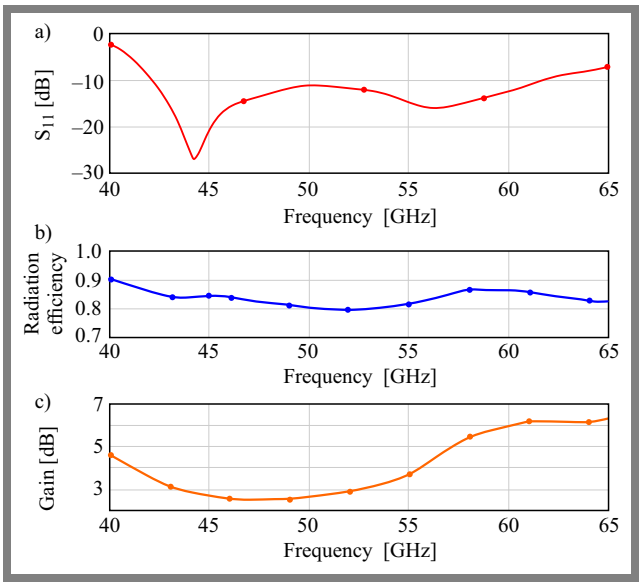

Fig. 2. a) S_{11} of the single antenna element, b) radiation efficiency, and c) gain.

The two-line steps have dimensions of 0.65 × 3.62 mm and 1.40 × 4.60 mm, respectively, while the symmetric ladder steps measure 2 × 0.64 mm. Figure 1b shows the back view of the antenna element with an etched ground plane.

The performance of a single antenna element is evaluated in terms of resonance frequency, bandwidth, radiation patterns, gain, and efficiency. Figure 2a illustrates the results of simulations of the reflection coefficient S_{11} versus frequency. It is clear that S_{11} remains below –10 dB across the entire frequency range of 42 to 62 GHz, i.e. ultra-wideband of 20 GHz with good impedance matching.

Figure 2b shows the radiation efficiency, which remains in the range of 80 – 88% within the entire working band. High antenna efficiency that remains stable within the ultra-wide working bandwidth is realized. The maximum gain reaches 6.2 dB, as can be observed in Fig. 2c. Such a high gain is required to eliminate the problem of propagating losses in

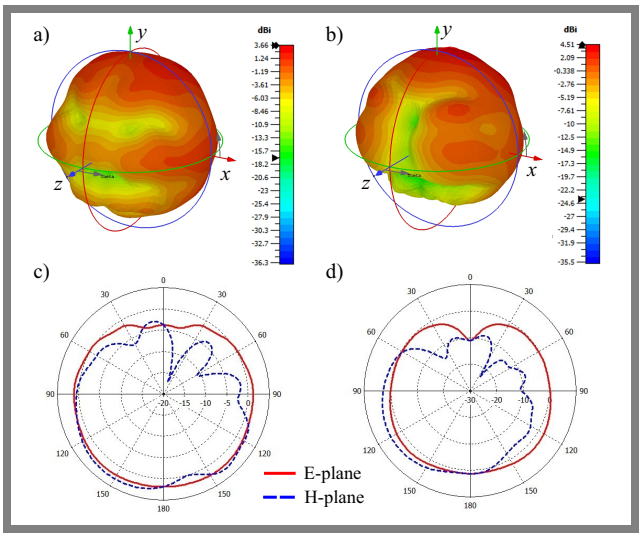

Fig. 3. 3D antenna radiation patterns at: a) 45 GHz, b) 55 GHz, and 2D antenna radiation patterns at: c) 45 GHz, d) 55 GHz.

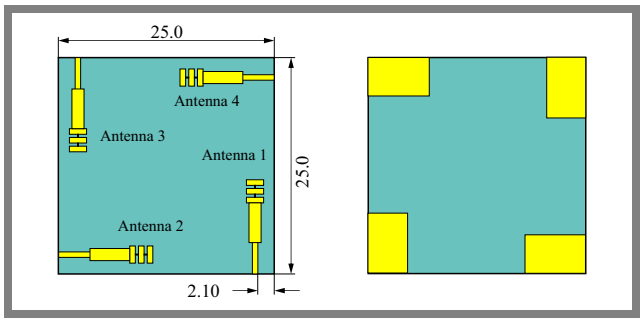

Fig. 4. Proposed antenna array system: a) top view and b) bottom view.

the mmWave spectrum. The radiation pattern for mmWave applications is visualized in Fig. 3. The three-dimensional radiation patterns at 45 GHz and 55 GHz are presented in Fig. 3a–b, while the E-plane and H-plane of the two-dimensional radiation patterns are illustrated in Fig. 3c–d.

4. Antenna Array System Design

Next, a four-element MIMO array is developed utilizing the single antenna described in Section 3. The MIMO elements are symmetrically and rotationally arranged in 90° intervals, creating a square configuration, as demonstrated in Fig. 4. Such a layout employs the polarization diversity approach used to minimize mutual coupling between the antennas. The dimensions of the MIMO system equal 25 × 25 mm.

Figure 5 illustrates the surface current distribution of the four-port MIMO antenna, where one port is excited and the other ports are connected to 50 Ω at the two frequencies of 45 GHz and 55 GHz, respectively. One may notice that the current flow is predominantly concentrated in the antenna unit with less propagation to other MIMO ports. The existence of multiple identical elements in a MIMO configuration results in an increase in mutual interference and ECC value between the antenna's elements. The orthogonal arrangement of the array results in the lowest possible mutual coupling.

Figure 6 shows the S-parameters for the proposed 4-element MIMO antenna design. All four return loss coefficients of the antennas are below –10 dB within the considered band of 20 from 42.3 to 62.3 GHz. This guarantees optimized impedance matching. Furthermore, a very slight variation can be observed between the return loss of the four elements and the single antenna element from Fig. 2. This proves a satisfactory mutual coupling between the antennas due

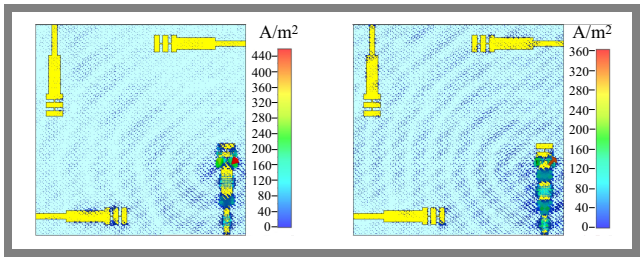

Fig. 5. Current distribution of the MIMO antenna array at: a) 45 GHz and b) 55 GHz.

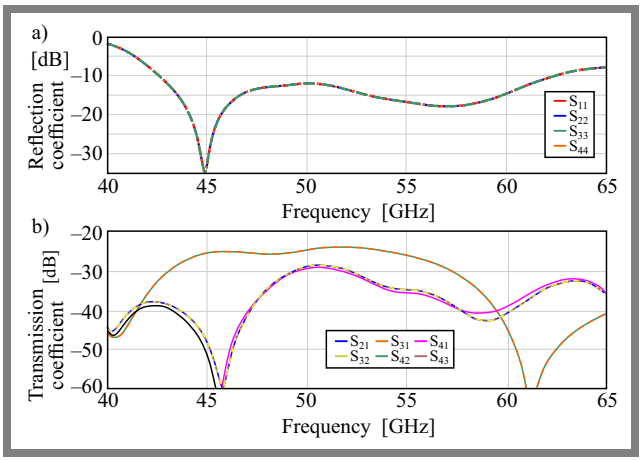

Fig. 6. S-parameters of the ultra-wideband quad MIMO antenna array: a) return loss and b) transmitting coefficients.

to their structural similarity and the orthogonal MIMO layout. All transmission coefficients are lower than –24 dB (Fig. 6b), i.e., high isolation is obtained by applying the polarization diversity technique.

Figure 7a shows the radiation efficiency and total antenna efficiency of the designed MIMO elements. The total antenna efficiencies for the overall operating band remain within the 66 to 72% range, while the antenna's radiation efficiencies vary from 78 to 80%. Figure 7b illustrates the gain of the proposed MIMO antennas in its operating band. The gain of all antennas is in range of 4.1 to 7.8 dB.

Figure 8 presents the two-dimensional radiation patterns at the frequency of 45 GHz and 55 GHz, respectively. As demonstrated, the peak gain of the antennas is acieved in diverse directions, demonstrating the highly desired benefit of the patterns' diversity. Furthermore, these radiation patterns fully cover all sides, proving excellent radiation coverage.

Analysis of diversity parameters such as ECC, MEG, and DG is as essential evaluation of the antenna's key performance parameters, including bandwidth, radiation patterns, resonance

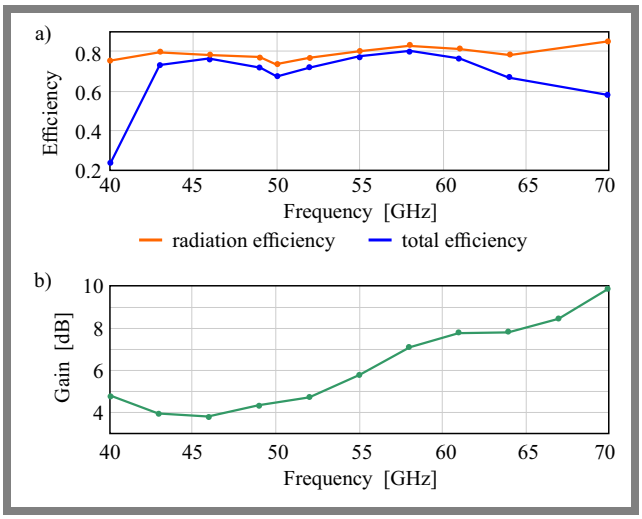

Fig. 7. Total and radiation efficiencies of the MIMO antenna array a) and gain b).

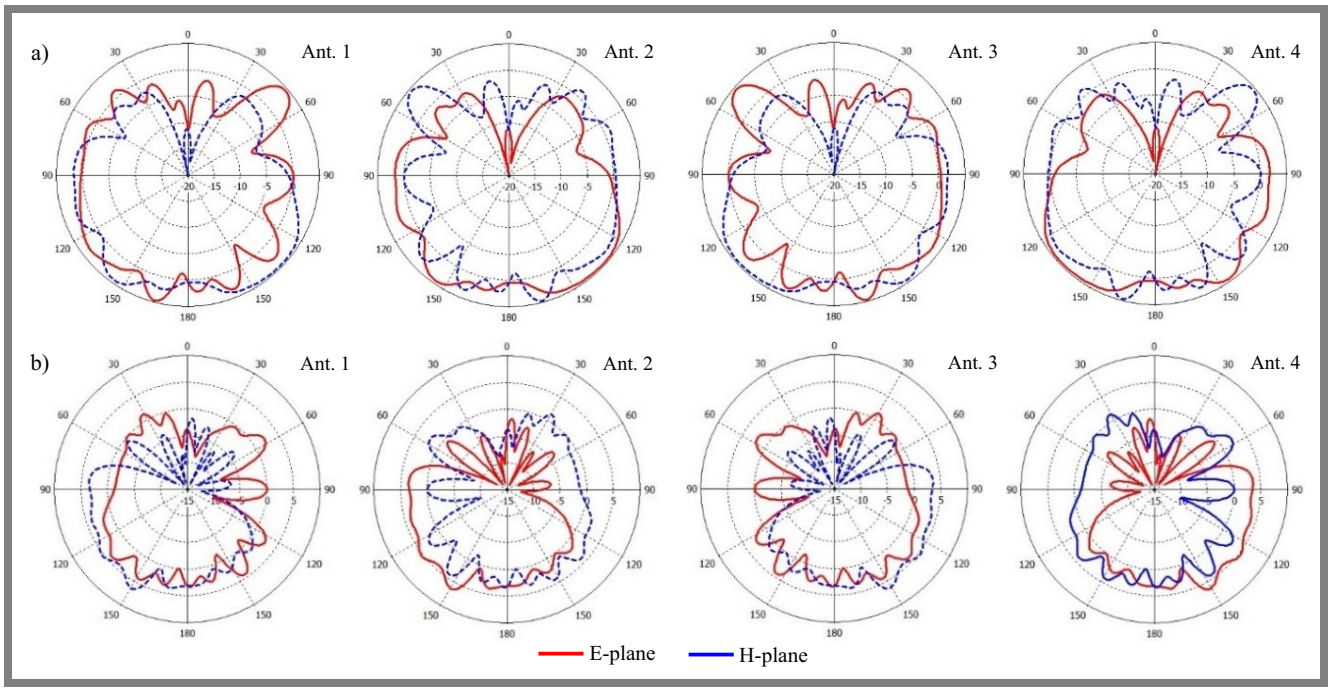

Fig. 8. 2D antenna radiation patterns at: a) 45 GHz and b) 55 GHz.

frequency, efficiency, and gain, as it allows to determine the efficacy of MIMO antenna arrays.

ECC determines the correlation between different antenna elements in a MIMO system. A low ECC value leads to less interdependence between the components and, hence, better MIMO diversity performance. Equations (1) and (2) illustrate how ECC is calculated using the scattering parameters and the far-field radiation pattern, respectively [13], [29].

$$ECC = |\rho_{ij}| = \frac{\left|S_{ii}^{*}S_{ij} + S_{ji}^{*}S_{jj}\right|^{2}}{\left(1-(|S_{ii}|^{2}+|S_{ji}|^{2})\right)\left(1-(|S_{jj}|^{2}+|S_{ij}|^{2})\right)}, \quad (1)$$

$$ECC = \frac{\left|\int\int_{4\pi}[\overrightarrow{F_1}(\theta,\varphi)^{*}\cdot\overrightarrow{F_2}(\theta,\varphi)]d\Omega\right|^{2}}{\int\int_{4\pi}\left|\overrightarrow{F_1}(\theta,\varphi)\right|^{2}d\Omega\cdot\int\int_{4\pi}\left|\overrightarrow{F_2}(\theta,\varphi)\right|^{2}d\Omega}, \quad (2)$$

where ρ_{ij} represents the envelope correlation coefficients (ECC) between the i and j antenna elements, $\overrightarrow{F_1}(\theta,\varphi)$ specifies 3D radiation pattern field with excitation at port i, $*$ signifies the Hermitian product and Ω indicates the solid angle.

In this work, the ECC from far field radiation patterns is taken into consideration. The expected ECC is less than 0.5, which falls within the permissible range for MIMO diversity. As illustrated in Fig. 9, very low ECCs are obtained, i.e. below 0.002, proving the very high diversity antenna system is achieved.

The MEG is an other important performance criterion for MIMO systems, defined as the ratio of the mean received power to the total mean incident power at the antenna [30]. MEG quantifies the mutual interaction of antenna elements

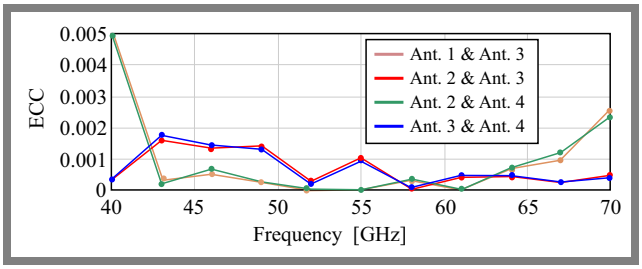

Fig. 9. ECCs of the proposed system.

and the statistical characteristics of the propagation environment. Equations (3) and (4) illustrate the method of obtaining the MEG value from s parameters or far-field radiations. The MIMO antennas MEGs must satisfy the requirements of Eq. (5). The XPR indicates the cross polarization power ratio, while the gains of the antenna are described by G_θ and G_φ. The P_θ and P_φ are incoming plane waves components. The variables i and k in Eq. (4) denote the observed antenna and the total antennas number, respectively. The MEGs for the i and j antennas are designated as MEG_i and MEG_j, respectively.

$$MEG_i = 0.5\left(1-\sum_{j=1}^{k}S_{ij}\right), \quad (3)$$

$$MEG = \int_0^{2\pi}\int_0^{\pi}\left(\frac{XPR}{1+XPR}G_\theta(\theta,\varphi)P_\theta(\theta,\varphi) + \frac{1}{1+XPR}G_\varphi(\theta,\varphi)P_\varphi(\theta,\varphi)\right)\sin\theta\, d\theta\, d\varphi, \quad (4)$$

$$\frac{MEG_i}{MEG_j} \cong 1. \quad (5)$$

Tab. 1. Comparison between the proposed MIMO array system and recent publications.

Ref.	Bandwidth [GHz]	Total efficiency [%]	Peak gain [dB]	Isolation [dB]	ECC	MIMO order	Overall size [mm]	Isolation technique
[15]	25.5 – 29.6	> 82	8.3	> 20	< 0.01	4 × 4	30 × 35	DGS
[20]	23 – 40	> 70	12	> 20	< 0.001	4 × 4	80 × 80	DGS
[32]	27.5 – 40	> 75	7.2	> 17	< 0.001	4 × 4	158 × 77.8	Spatial diversity and polarization diversity
[33]	20 – 32	80 – 90	6.5	> 20	< 0.001	4 × 4	24 × 32	Decoupling structure on ground plane
[34]	27 – 29	80 – 84	5.5	> 17	< 0.03	4 × 4	30 × 30	Polarization diversity
This work	42.3 – 63.3	66 – 72	7.8	> 24	< 0.002	4 × 4	25 × 25	Polarization diversity

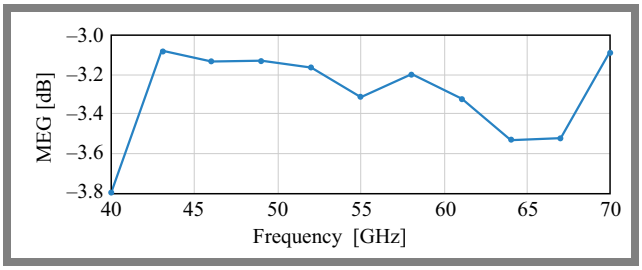

Fig. 10. MEGs of the antenna system.

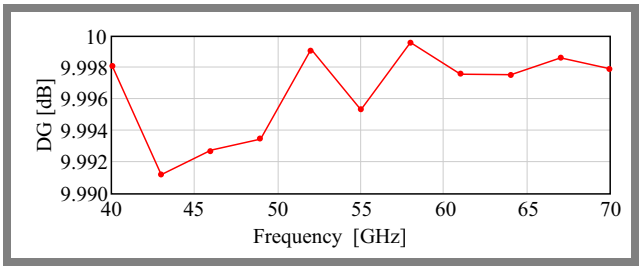

Fig. 11. DG for the proposed MIMO system.

Figure 10 illustrates the MEGs for the four antennas according to Eq. (3), on the far-field radiation, assuming a Gaussian distribution in the elevation direction and a uniform distribution in the azimuth direction. One may notice that the proposed MIMO antenna array meets the condition from Eq. (5) and the MEGs of the quad antennas maintain stable across the frequency band of interest.

The DG value is derived from ECC as [30]:

$$G_{DG} = 10 \times \sqrt{1 - |\rho|^2}\,. \tag{6}$$

A high diversity gain value signifies exceptional performance, while better isolation between the antenna's elements correlates with higher diversity gain values. Figure 11 shows the diversity gain of a MIMO antenna system determined using Eq. (6), which is greater than 9.99 dB across the working band, and thus demonstrates the effective diversity performance of the proposed design.

A comparison between the proposed MIMO array system and other similar recent publications is shown in Tab. 1. The comparison covers several aspects, including bandwidth, efficiency, gain, isolation between antenna elements, ECC, MIMO order, size, and the isolation technique used. The proposed MIMO system features compact dimensions, wide bandwidth, high gain, and decent isolation. The results indicate that the designed MIMO antenna system is a promising candidate for future mmWave devices.

5. Conclusions

This study presents the design of the four-port MIMO antenna array working in the ultra-wideband range from 42.3 to 63.3 GHz. The proposed antenna demonstrates stable radiation properties throughout the operating band, with a peak gain of 7.8 dB. Due to adopting the orthogonality polarization diversity approach, high isolation values of over 24 dB are achieved. Very low ECCs below 0.002 and high DGs over 9.998 dB are obtained as well, ensuring high diversity performance.

Furthermore, optimal MEG values are achieved and the criteria for good performance of the MIMO antenna system are validated by simulations performed using the CST Microwave Studio software. The antenna has achieved the desired performance in terms of gain, efficiency, and radiation properties.

References

[1] M. Shafi *et al.*, "5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice", *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017 (https://doi.org/10.1109/JSAC.2017.2692307).

[2] M. Agiwal, H. Kwon, S. Park, and H. Jin, "A Survey on 4G-5G Dual Connectivity: Road to 5G Implementation", *IEEE Access*, vol. 9, pp. 16193–16210, 2021 (https://doi.org/10.1109/ACCESS.2021.3052462).

[3] J. Cheng, Y. Yang, X. Zou, and Y. Zuo, "5G in Manufacturing: A Literature Review and Future Research", *The International Journal of Advanced Manufacturing Technology*, vol. 131, no. 11, pp. 5637–5659, 2024 (https://doi.org/10.1007/s00170-022-08990-y).

[4] B.C. Tedeschini, M. Nicoli, and M.Z. Win, "On the Latent Space of mmWave MIMO Channels for NLOS Identification in 5G-advanced Systems", *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 6, pp. 1655–1669, 2023 (https://doi.org/10.1109/JSAC.2023.3273769).

[5] W.A.E. Ali, A.A. Ibrahim, and A.E. Ahmed, "Dual-band Millimeter Wave 2×2 MIMO Slot Antenna with Low Mutual Coupling for 5G Networks", *Wireless Personal Communications*, vol. 129, no. 4, pp. 2959–2976, 2023 (https://doi.org/10.1007/s11277-023-10267-w).

[6] W.T. Sethi *et al.*, "Pattern Diversity Based Four-element Dual-band MIMO Patch Antenna for 5G mmWave Communication Networks", *Journal of Infrared, Millimeter, and Terahertz Waves*, vol. 45, no. 5, pp. 521–537, 2024 (https://doi.org/10.1007/s10762-024-00983-0).

[7] S. Kumar *et al.*, "Fifth Generation Antennas: A Comprehensive Review of Design and Performance Enhancement Techniques", *IEEE Access*, vol. 8, pp. 163568–163593, 2020 (https://doi.org/10.1109/ACCESS.2020.3020952).

[8] A.S. Dixit and S. Kumar, "A Survey of Performance Enhancement Techniques of Antipodal Vivaldi Antenna", *IEEE Access*, vol. 8, pp. 45774–45796, 2020 (https://doi.org/10.1109/ACCESS.2020.2977167).

[9] S.F. Farida, P.M. Hadalgi, P.V. Hunagund, and S.R. Ara, "Effect of Substrate Thickness and Permittivity on the Characteristics of Rectangular Microstrip Antenna", *1998 Conference on Precision Electromagnetic Measurements Digest*, Washington, USA, 1998 (https://doi.org/10.1109/CPEM.1998.700074).

[10] M.M. Honari, M.S. Ghaffarian, P. Mousavi, and K. Sarabandi, "A Wideband High-gain Planar Corrugated Antenna", *2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, Montreal, Canada, 2020 (https://doi.org/10.1109/IEEECONF35879.2020.9330020).

[11] Z.X. Wang and W.B. Dou, "Dielectric Lens Antennas Designed for Millimeter Wave Application", *2006 Joint 31st International Conference on Infrared Millimeter Waves and 14th International Conference on Terahertz Electronics*, Shanghai, China, 2006 (https://doi.org/10.1109/ICIMW.2006.368584).

[12] Z. Zhang *et al.*, "Dual-band Focused Transmitarray Antenna for Microwave Measurements", *IEEE Access*, vol. 8, pp. 100337–100345, 2020 (https://doi.org/10.1109/ACCESS.2020.2998131).

[13] M.Y. Muhsin, A.J. Salim, and J.K. Ali. "An Eight-element Multi-band MIMO Antenna System for 5G Mobile Terminals", *AIP Conference Proceedings*, vol. 2651, no. 1, 2023 (https://doi.org/10.1063/5.0105773).

[14] A. Abdelraheem, H. Elregaily, A.A. Mitkees, and M. Abdalla, "A Hybrid Isolation in Two-element Directive UWB MIMO Antenna", *IETE Journal of Research*, vol. 69, no. 1, pp. 499–508, 2023 (https://doi.org/10.1080/03772063.2020.1830863).

[15] M. Khalid *et al.*, "4-Port MIMO Antenna with Defected Ground Structure for 5G Millimeter Wave Applications", *Electronics*, vol. 9, no. 1, art. no. 71, 2020 (https://doi.org/10.3390/electronics9010071).

[16] N. Yoon and C. Seo, "A 28-GHz Wideband 2×2 U-slot Patch Array Antenna", *Journal of Electromagnetic Engineering and Science*, vol. 17, no. 3, pp. 133–137, 2017 (https://doi.org/10.5515/JKIEES.2017.17.3.133).

[17] R. Anitha *et al.*, "A Compact Quad Element Slotted Ground Wideband Antenna for MIMO Applications", *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 10, pp. 4550–4553, 2016 (https://doi.org/10.1109/TAP.2016.2593932).

[18] C.R. Jetti *et al.*, "Design and Analysis of Modified U-shaped Four Element MIMO Antenna for Dual-band 5G Millimeter Wave Applications", *Micromachines*, vol. 14, no. 8, art. no. 1545, 2023 (https://doi.org/10.3390/mi14081545).

[19] D.A. Sehrai *et al.*, "A Novel High Gain Wideband MIMO Antenna for 5G Millimeter Wave Applications", *Electronics*, vol. 9, no. 6, art. no. 1031, 2020 (https://doi.org/10.3390/electronics9061031).

[20] E. Al Abbas, M. Ikram, A.T. Mobashsher, and A. Abbosh, "MIMO Antenna System for Multi-band Millimeter-wave 5G and Wideband 4G Mobile Communications", *IEEE Access*, vol. 7, pp. 181916–181923, 2019 (https://doi.org/10.1109/ACCESS.2019.2958897).

[21] F.W. Ardianto, F.F. Lanang, S. Renaldy, and T. Yunita, "Design MIMO Antenna with U-Slot Rectangular Patch Array for 5G Applications", *2018 International Symposium on Antennas and Propagation (ISAP)*, Busan, South Korea, 2018.

[22] Y. Rahayu and I.R. Mustofa, "Design of 2×2 MIMO Microstrip Antenna Rectangular Patch Array for 5G Wireless Communication Network", *2017 Progress in Electromagnetics Research Symposium-Fall (PIERS–FALL)*, Singapore, 2017 (https://doi.org/10.1109/PIERS-FALL.2017.8293591).

[23] M.K. Ishfaq, T.A. Rahman, Y. Yamada, and K. Sakakibara, "8×8 Phased Series Fed Patch Antenna Array at 28 GHz for 5G Mobile Base Station Antennas", *2017 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC)*, Verona, Italy, 2017 (https://doi.org/10.1109/APWC.2017.8062268).

[24] Y. Wang and D. Piao, "A Compact Size Dual-polarized High-gain Resonant Cavity Antenna at 28 GHz", *2017 International Applied Computational Electromagnetics Society Symposium (ACES)*, Suzhou, China, 2017.

[25] A. Thatere, P.L. Zade, and D. Arya, "Bandwidth Enhancement of Microstrip Patch Antenna Using 'U' Slot with Modified Ground Plane", *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, Bhubaneswar, India, 2015 (https://doi.org/10.1109/ICMOCE.2015.7489779).

[26] I. Rosaline, A. Kumar, P. Upadhyay, and A.H. Murshed, "Four Element MIMO Antenna Systems with Decoupling Lines for High-speed 5G Wireless Data Communication", *International Journal on Antennas and Propagation*, vol. 2022, no. 1, art. no. 9078929, 2022 (https://doi.org/10.1155/2022/9078929).

[27] K.S. Sultan and H.H. Abdullah, "Planar UWB MIMO-diversity Antenna with Dual Notch Characteristics", *Progress in Electromagnetics Research C*, vol. 93, pp. 119–129, 2019 (https://doi.org/10.2528/PIERC19031202).

[28] R.E.A. Shehata, A. Elboushi, M. Hindy, and H. Elmekati, "Metamaterial Inspired LPDA MIMO array for upper band 5G applications", *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 32, no. 8, art. no. 23212, 2022 (https://doi.org/10.1002/mmce.23212).

[29] M.Y. Muhsin, A.J. Salim, and J.K. Ali, "Compact MIMO Antenna Designs Based on Hybrid Fractal Geometry for 5G Smartphone Applications", *Progress in Electromagnetics Research C*, vol. 118, pp. 247–262, 2022 (https://doi.org/10.2528/PIERC22012808).

[30] Z.F. Al-Azzawi *et al.*, "Designing Eight-port Antenna Array for Multi-band MIMO Applications in 5G Smartphones", *Journal of Telecommunications and Information Technology*, no. 4, pp. 18–24, 2023 (https://doi.org/10.26636/jtit.2023.4.1297).

[31] M.Y. Muhsin *et al.*, "Isolation Techniques in MIMO Antennas for 5G Mobile Devices (Comprehensive Review)", *Radioelectronics and Communications Systems*, vol. 66, no. 6, pp. 263–287, 2023 (https://doi.org/10.3103/S0735272723040027).

[32] M. Bilal *et al.*, "High-isolation MIMO Antenna for 5G Millimeter-wave Communication Systems", *Electronics*, vol. 11, no. 6, art. no. 962, 2022 (https://doi.org/10.3390/electronics11060962).

[33] H. Elmannai *et al.*, "Design and Characterization of a Meandered V-shaped Antenna Using Characteristics Mode Analysis and its MIMO Configuration for Future mmWave Devices", *AEU – International Journal of Electronics and Communications*, vol. 186, art. no. 155477, 2024 (https://doi.org/10.1016/j.aeue.2024.155477).

[34] S. Rahman *et al.*, "Nature Inspired MIMO Antenna System for Future mmWave Technologies", *Micromachines*, vol. 11, no. 12, art. no. 1083, 2020 (https://doi.org/10.3390/mi11121083).

Muhannad Y. Muhsin, Ph.D.
Department of Electrical Engineering
https://orcid.org/0000-0003-3937-4467
E-mail: muhannad.y.muhsin@uotechnology.edu.iq
University of Technology – Iraq, Baghdad, Iraq
https://uotechnology.edu.iq

Zainab S. Muqdad, M.Sc.
Electrical Engineering Department
https://orcid.org/0009-0003-1342-7994
E-mail: zainab.salam@uomustansiriyah.edu.iq
Mustansiriyah University, Baghdad, Iraq
https://uomustansiriyah.edu.iq

Asaad H. Sahar, Ph.D.
Electronics and Communication Engineering Department
https://orcid.org/0000-0003-3655-0162
E-mail: asaad.h@coeng.uobaghdad.edu.iq
University of Baghdad, Baghdad, Iraq
https://en.uobaghdad.edu.iq

Zainab F. Mohammad, M.Sc.
Communication Engineering Department
https://orcid.org/0000-0002-1627-751X
E-mail: zainab.f.mohammad@uotechnology.edu.iq
University of Technology – Iraq, Baghdad, Iraq
https://uotechnology.edu.iq

Hussam AL-Saedi, Ph.D.
Communication Engineering Department
https://orcid.org/0000-0002-2029-7361
E-mail: hussam.h.ali@uotechnology.edu.iq
University of Technology – Iraq, Baghdad, Iraq
https://uotechnology.edu.iq

Compressive Sensing-based Differential Channel Feedback Scheme Using Subspace Matching Pursuit Algorithm for B5G Wireless Systems

Baranidharan V[1,2] and Surendar M[1]

[1]*National Institute of Technology Puducherry, Karaikal, India,*
[2]*Bannari Amman Institute of Technology, Sathy, India*

https://doi.org/10.26636/jtit.2025.1.1904

Abstract — Millimeter wave (mmWave) massive multi-input multi-output (MIMO) systems are the promising technology for next-generation 5G wireless systems and beyond. Sparse signal recovery and channel feedback are challenging and fundamental problems affecting downlink transmission due to the substantial increase in channel matrix size in mmWave systems. To overcome the overhead of the channel and improve CS recovery effectiveness, this article proposes the joint use of the subspace matching search algorithm and differential operation for channel impulse response (CIR). Here, the current CIR is converted to a differential CIR using operations between the current and previous CIRs. The differential CIR is then compressed and fed back to the base station. Subsequently, this differential CIR is recovered using the subspace matching search algorithm. Such a scheme leverages effective structural sparsity through a combination of subspace and differential operations. The adaptive algorithm adaptively selects relevant subspaces based on coefficients. The simulation results show that the proposed scheme reduces channel overhead by 36% and 24% at compression ratios of 25% and 45%, respectively, over different time slots in mmWave massive MIMO systems.

Keywords — channel impulse response, channel state information, compressive sensing, mmWave

1. Introduction

Massive millimeter MIMO systems are a key technology for 5G communication systems and beyond [1]. Such systems are equipped with many directional antennas to achieve effective beamforming gains [2] required to cope with higher path losses. To overcome these issues, effective beamforming strategies are proposed along with a large number of antenna arrays. There is always a trade-off between performance and system complexity when using such beamforming strategies [3]. Effective channel state information (CSI) estimation is very important to obtain optimal performance of any scheme.

For time division duplexing (TDD) scheme-based mmWave massive MIMO systems, the downlink (DL) channel CSI is estimated immediately by DL using received the transmitted pilot signals due to its channel reciprocity. However, in frequency division duplexing (FDD) systems, due to non-existence of channel reciprocity [4], the uplink channel (UL) CSI is not always equivalent to the DL CSI. Excessive channel feedback is always required to estimate CSI, making channel feedback very challenging [5]. To address this issue, orthogonal pilot signals are transmitted. However, the use of such orthogonal pilots results in increased resource utilization within the communication system.

Generally, the estimation of mmWave channels always exhibits various sparse scattering characteristics [6]. The mmWave channel matrix always exploits a highly sparse channel matrix with non-zero elements based on the positions of angle of arrival (AoA) and angle of departure (AoD) of the various dominant multipath signals [7]. In massive MIMO systems, effective recovery of the sparse signal is considered as the mmWave-based channel estimation problem.

Compressive sensing (CS) algorithms are an effective approach [8] adopted to reconstruct undetermined linear systems. MmWave channel estimation is generally formulated as a sparse signal recovery problem, which provides a compressive measurement that combines the effects of analog and digital precoders and combiners. CS-based algorithms are introduced to recover these sparse signal recovery problems by effectively quantifying AoA and AoD of various multipath signals for the formation of uniform grids. Some hierarchical codebook-based schemes are designed in such a way as to estimate the millimeter wave path parameters of the channels in MIMO systems.

For CS recovery, the orthogonal matching pursuit (OMP) algorithm is employed to quantize non-uniform angle grids. Based on this evidence, direct estimation of all the entries in the mmWave channel of massive MIMO systems is a challenging task. CS-based channel feedback schemes [9] are proposed to exploit the antenna's spatial correlations by compressing the channel matrix of mmWave channels, thereby reducing the feedback overhead.

The channel impulse response (CIR) is directly compressed and recovered using effective CS algorithms, exploiting them in the time domain to reduce feedback overhead [10]. In

mmWave channels, the overhead is very high because of the sparsity nature of the channel impulse response.

2. Related Works

Many CS-based recovery algorithms have been analyzed and proposed in recent studies to reduce channel feedback. The sparse recovery problem [11] is formulated based on the multipath signal by quantizing the angle of arrivals (AoA) and angle of departures (AoDs) into uniform grids. Some effective and adaptive CS-based algorithms are developed to estimate channel parameters. To facilitate this estimation operation, multiresolution codebook-based CS algorithms have been developed [12]. Some methods improve channel recovery performance by identifying non-uniformly quantized angles. The approach presented in [13] also reduces coherence redundancy of the systems. To find the array response vectors and to estimate non-uniform quantized angle grids, the orthogonal matching pursuit (OMP) algorithm is proposed [14].

Other basic search algorithms are also introduced to reduce computational complexity of existing systems [15]. In [16], two different adaptive low-complexity CS algorithms are presented which iteratively estimate channel parameters. A block sparsity-based CS algorithm is proposed in [17], considering multi-user mmWave massive MIMO systems.

In studies [18]–[20], CS algorithms are used for narrowband frequency-flat channels. In such cases, they were analyzed in the time domain instead of the angular or frequency domains. The main challenge in estimating channels in frequency-selective channels is the need for common supports between the mmWave channel sparsity matrices, which inevitably leads to a better trade-off between complexity and performance. In [20], uplink frequency-selective channels are estimated using a two-stage CS algorithm, with precoders and combiners. The major challenge is that, in the time domain, wideband mmWave channels exploit delays in sparse channel vectors. To address the problem of sparse channel recovery, hybrid architectures are proposed, but, unfortunately, they generally require limited training [21].

In paper [22], a two-step time domain estimator based on OMP and the least squares methods is formulated to reduce computational complexity. The sparsity nature of mmWave systems in the angle and delay domains is defined jointly, and the problem is effectively iterated by the CS-based message passing method (MPM) [23] which exploits its structured sparsity to find the nearest neighbor pattern [24], [25]. In addition to exploiting the sparsity nature, other structured channel models, such as low rank [26], uniform grid structure [27], [28], and jointly sparse and low rank structures [29], are also used and estimated by leveraging the sparsity feature. The sparse recovery problem is defined for both time and frequency domains. Different CS algorithms [30] are used for recovery in the time domain the frequency domain, or a combination of both, as proposed in [31]–[33].

Most of the existing works, e.g. [34] and [35], dealing with mmWave channel estimation techniques, focus only on narrowband frequency models. To provide context, techniques focusing on wideband mmWave channels and frequency-selective channels are also analyzed in both frequency and time domains [36]. To analyze downlink frequency-selective channels, CS techniques will be helpful in finding the common support which exhibits the sparsity of mmWave channels in the frequency domain [20].

Other methods use effective precoders and combiners employing both on-grid and off-grid techniques [37]. In general, these wideband mmWave channel estimation techniques widely exploit delay sparsity while using small training overheads. Lastly, two-step channel estimation methods are proposed using least squares estimation along with orthogonal matching search algorithms for CS recovery [38].

In this paper, we propose a modified compressive sensing-based differential channel feedback scheme using the subspace matching pursuit recovery algorithm (SMP-DF CS) to reduce feedback overhead and improve the performance of mmWave massive MIMO systems. The said scheme exploits the temporal correlation of highly time-varying mmWave massive MIMO systems. This temporal correlation property exists in both distributed and centralized systems.

The proposed algorithm effectively reduces the feedback overhead and computational complexity at different compressive ratios and demonstrates improved performance for the differential feedback scheme of mmWave massive MIMO systems.

The main contribution of this article is summarized as follows:

- A joint framework is proposed for a modified SMP-based CS recovery scheme with differential operation for the effective estimation of the channel impulse response in the mmWave channel. This concept effectively leverages the sparsity nature of mmWave channels in the angular domain and reduces channel overhead.
- The CS recovery scheme introduced uses a modified subspace matching pursuit algorithm to improve CIR recovery by effectively searching subspaces in each iteration in order to form the support vectors.
- The proposed adaptive algorithm selects relevant subspaces based on coefficients, rather than choosing each basis for iterations, thus resulting in faster convergence and a better achievable sum rate.

The paper is organized as follows. Section 3 explains the proposed CS-based differential channel feedback scheme for subspace matching using a model of the system model and relevant preliminaries. Section 4 gives a detailed explanation of the performance of NMSE in different SNR regimes. Conclusions are presented in Section 5.

3. Proposed SMP-DF CS Scheme

3.1. System Model

The model of a mmWave massive MIMO system is equipped with N_t and N_R transmitting and receiving antennas, respectively. The received downlink signal transmission r_p, based

on the training symbol, is expressed as:

$$r_p = H_p f_p s_p + n_p \,, \tag{1}$$

where H_p is the channel matrix of the downlink mmWave system, f_p is the vector representing beamforming, s_p denotes the received symbol vector, and n_p represents the complex white noise vector with Gaussian distribution. The combined vectors formed to detect the transmitted symbols are given by:

$$y_p = W^H H_p f_p s_p + W^H n_p \,, \tag{2}$$

where vector $w = [W_1, W_2, W_3, \ldots, W_{Nq}]$.

The downlink channel matrix equivalent to its transmitted pilots is given as $H_p = H$ and is based on block fading. The received signal vector is formulated as:

$$Y = [y_1, y_2, y_3, \ldots, y_{N_p}] \,. \tag{3}$$

After receiving the pilot signals, Eq. (3) becomes:

$$Y = W^H HFS + W^H N \,, \tag{4}$$

where s represents the diagonal matrix.

The mmWave signal channel is modelled as:

$$H = \sqrt{\frac{N_T N_R}{L}} \sum_{l=1}^{L} \alpha_l \,.\, a_r(\theta_l) \,.\, a_t^H(\phi_l) \,, \tag{5}$$

where L represents the number of scatters, θ_l and φ_l denote the array response vectors that are associated with azimuth and elevation angles, respectively.

The array response vectors $a_t(\theta_l)$ of the transmitter are:

$$\sqrt{\frac{1}{N_t}} \left[1, e^{j\frac{2\pi}{\lambda} d \sin(\theta_l)}, \ldots, e^{j\frac{2\pi}{\lambda}(N_t-1) d \sin(\theta_l)}\right] . \tag{6}$$

The array response vectors $a_r(\varphi_l)$ of the receiver are formulated as:

$$\sqrt{\frac{1}{N_r}} \left[1, e^{j\frac{2\pi}{\lambda} d \sin(\phi_l)}, \ldots, e^{j\frac{2\pi}{\lambda}(N_r-1) d \sin(\phi_l)}\right] , \tag{7}$$

where d and λ represent the distance between the adjacent antennas in the UPA and the wavelength of the signal, respectively.

The model of the system is finally defined in a compact form as:

$$H = A_r \,.\, H_\alpha \,.\, A_t^H \,, \tag{8}$$

where the terms $A_t = [a_t(\theta_1), a_t(\theta_2), \ldots, a_t(\theta_l)]$ and $A_r = [a_r(\phi_1), a_r(\phi_2), \ldots, a_r(\phi_l)]$ represents the steering matrix of transmitter and receivers of this systems, respectively.

The channel matrix is formulated as follows:

$$H_\alpha = \sqrt{\frac{N_t \,.\, N_r}{L}} \operatorname{diag}\left[\alpha_1, \alpha_2, \ldots, \alpha_L\right] , \tag{9}$$

where H_α is the diagonal matrix based on the steering vectors of the transmitted and received signals.

3.2. Time Varying mmWave Massive MIMO Channel

The temporal time-varying channel impulse response of the mmWave massive MIMO system of the t-th time slot of the n-th transmitting antenna at the base station (BS) is considered. The effective channel model for the received signal at a single-antenna user is given as:

$$h(t)_n = \left[h_n^{(t)}(0), h_n^{(t)}(1), \ldots, h_n^{(t)}(L-1)\right] , \tag{10}$$

where N is the total number of transmitting antennas and L is the maximum channel delay spread.

The channel impulse response (CIR) is always very sparse because of its nature in mmWave communication, as it typically consists of only a few dominant propagation paths. These paths play a significant role in improving channel response.

The sparsity nature of the CIR series $\left[h(t)_n^T\right]_{t=1}^T$ consists of T consecutive time slots that exhibit high temporal correlation values even in massive MIMO channels with fast time-varying MIMO channels. The change in temporal correlations always exists through the support vectors which refer to the position of non-zero elements and their amplitudes.

The time-varying and sparse nature of the CIR equation is formulated by the support vector $p(t)_n$ and the amplitude vectors $a(t)_n$. Then the Eq. (10) becomes:

$$h(t)_n = a(t)_n \circ p(t)_n \,, \tag{11}$$

where $p(t)_n$ is the support vector and $a(t)_n$ is the amplitude vector at time slot t of the n-th transmitting antenna. The "$\circ$" symbol represents the Hadamard product which denotes element-wise multiplication.

In order to model the rapid variations of this mmWave channel, support vectors $p(t)_n$ over time slot t in l elements can be represented as a first-order Markov process. This Markov process is characterized on two different transition probabilities, denoted as P_{01} and P_{10}. These terms are distributed $m(1)_n$ at the initial time slot $t = 1$. For any steady-state in the Markov process, the transition probabilities for all values of t and n are given as:

$$\Pr\left[p(t)_n(l) = 1\right] = m \,. \tag{12}$$

The other transmission probabilities P_{10} are represented as:

$$P_{10} = \frac{\mu \, P_{01}}{1 - \mu} \,. \tag{13}$$

CIR amplitude over the time slot t is modeled using a first-order autoregressive model, given as:

$$a(t)_n = \rho \,.\, a(t-1)_n + \sqrt{1 - \rho^2 \, w(t)} \,, \tag{14}$$

where r represents the correlation coefficient, ρ is the zero-order Bessel function, f_d represents the maximum Doppler frequency, τ is the duration between the time intervals, and $w(t)$ stands for independent noise vectors, with all elements always assumed to be independent and identically distributed (iid) with a normal distribution.

3.3. SMP-DF CS Scheme

The proposed algorithm (depicted in Fig. 1) enables the adaptation of beamforming techniques and provides effective reduction of channel feedback, as well as improves robustness to channel estimation errors. This scheme directly compresses

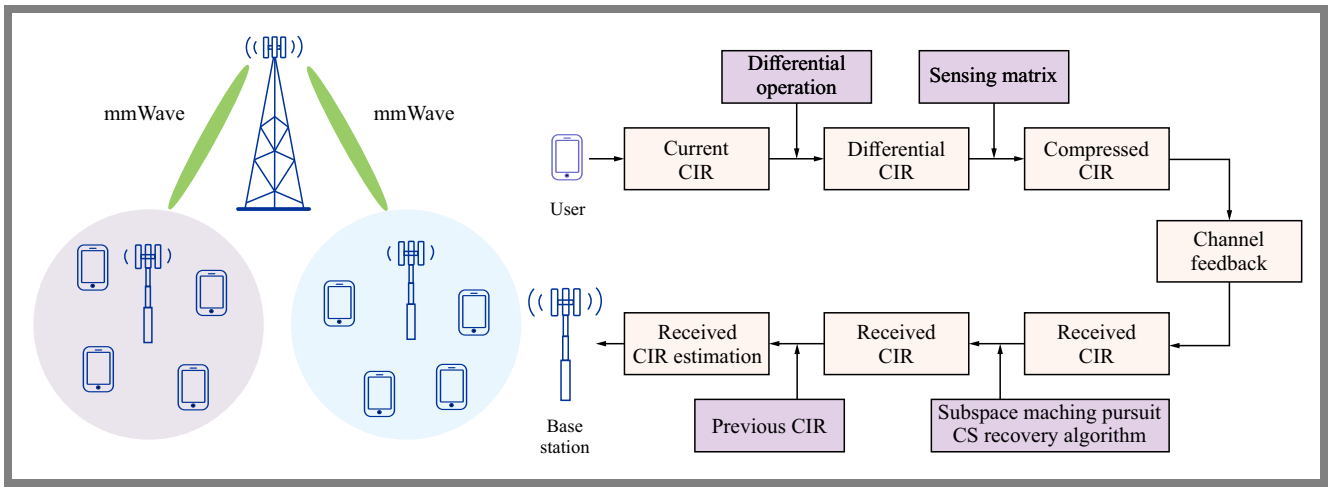

Fig. 1. Subspace matching pursuit CS recovery-based differential channel feedback.

sparse CIR using the sensing matrix based on the CS algorithm.

Before compression, the sparse CIR is converted into a differential CIR by using differential operations between the current and previous CIRs. This differential CIR has better sparsity than the original CIR. It computes the difference between the estimated $h_n(t-1)$ at $(t-1)$ slot and the previous CIR $h(t)$ at t slot. The differential CIR is expressed as [9]:

$$\triangle h_n^{(t)} = h_n^{(t)} - h_n^{(t-1)} \,. \tag{15}$$

Algorithm 1 Subspace matching pursuit CS algorithm

Input: measurement vector z, measurement matrix Φ, sparsity level s, maximum number of iterations max_iter

Output: recovered signal a, number of used iterations $used\_iter$

Start

$d \leftarrow$ dimension of Φ
$a \leftarrow \mathbf{0}_d$ ▷ Initialize recovered signal
$\rho \leftarrow z$ ▷ Initialize residual
$T \leftarrow \emptyset$ ▷ Initialize support
for $it = 1$ to max_iter **do**
  Compute inner products: $inner\_products \leftarrow |\Phi^T \rho|$
  Update support: $T \leftarrow$ indices of top $2s$ elements of $inner\_products$
  Estimate signal: $b \leftarrow \mathbf{0}_d$, $b_T \leftarrow (\Phi(:,T))^+ z$
  Prune signal estimate: keep top s coefficients; $T \leftarrow$ indices of top s elements of $|b|$
  Update recovered signal: $a \leftarrow \mathbf{0}_d$, $a_T \leftarrow b_T$
  Update residual: $r \leftarrow z - \Phi a$
  if $\|\rho\| < 1\mathrm{E}-3\,\|z\|$ **then**
    break
  end if
end for
$used\_iter \leftarrow it$

End

After substituting the current and previous CIRs according to Eq. (14), the equation becomes:

$$\triangle h_n^{(t)} = p_n^{(t)} \circ \left[\sqrt{(1-\rho^2 w(t)} - (1-\rho) a_n^{(t-1)}\right] + \left[(p_n^{(t)} - p_n^{(t-1)} \circ a_n^{(t-1)}\right] \,. \tag{16}$$

The first term of Eq. (16), which consists of the CIR amplitude, is almost negligible. Similarly, the second term, representing the non-zero elements, is also very small. To avoid errors in feedback propagation, mobile users initialize the CIR based on a high and effective compression ratio to enable precise recovery at the base station.

After compression of the differential CIR, the channel is fed back to the BS. On the BS side, the proposed scheme is based on three important steps. The received CIR is efficiently recovered using the subspace matching pursuit (SMP)-based CS algorithm.

Based on the compressive sensing (CS) theory, the highly sparse differential CIR is compressed using the sensing matrix. Then, the measurement vector is given as:

$$y = \phi \,.\, \triangle h_n^{(t)} \,, \tag{17}$$

where ϕ represents the sensing matrix and $\triangle h_n^{(t)}$ represents the differential CIR.

On the receiving side of the BS, measurement vector y is fed back through the channel:

$$y = \phi \,.\, \triangle h_n^{(t)} + n \,, \tag{18}$$

where n represents the noise of the channel. Here, the entities are also considered i.i.d. and follow a normal distribution.

On the BS side, the noise vector is added to the measurement vector. The SMP-based CS recovery algorithm is adopted to recover differential CIRs. The SMP-based CS recovery algorithm iteratively refines the differential CIR by updating its support and amplitudes based on the received measurement vector. This Algorithm 1 initializes such parameters as d, which represents the dimension of the sensing matrix, a which is a zero vector, ρ – denoting the measurement vector, and T, which represents the support vector of the signal.

Tab. 1. Simulation parameters for the proposed scheme.

Parameter	Value
Measurement matrix dimensions L	200
Transmission probability at initial time slot ρ	0.05
Initial probability of support vector m	0.1
Maximum Doppler frequency f_d	10 Hz
Time slot duration τ	1 ms
Standard deviation of independent noise vector σ	1
Number of antennas N	32

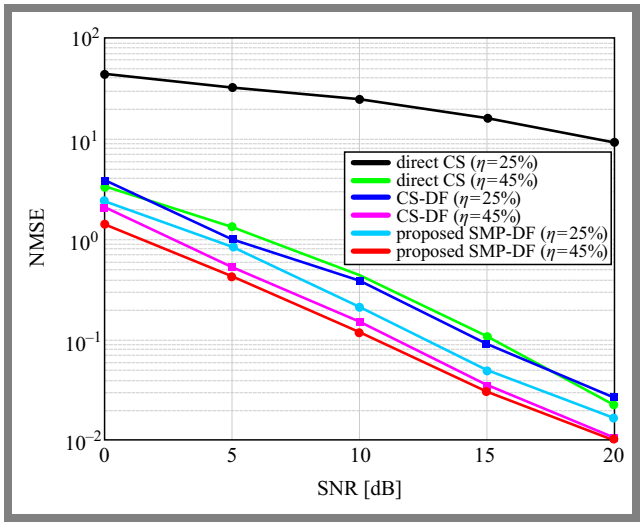

Fig. 2. Comparison of NMSE versus SNR for the proposed scheme and existing CS schemes (η =25% and 45%).

In each iteration stage of this algorithm, the inner products between the residual and columns are computed first to find the dimension of the sensing matrix. Thereafter, the support vector is selected based on the magnitudes of the first two elements. The signal is estimated by optimizing the least squares problem. The signal is estimated based on the coefficient with the highest magnitude. The residue is then updated by calculating the difference between the updated signal and its measurements.

4. Simulation Results

This section presents the numerical results to illustrate the proposed channel estimation algorithm and other existing methods. Here, we have considered the mmWave system architecture where both the transmitter and receiver are equipped with uniform planar arrays. Half of the wavelength of the signal $\frac{\lambda}{2}$ will be considered as the distance between the adjusted antenna elements. Similarly, the values of AoA and AoD of each path are selected using a uniform distribution form $[0, 2\pi]$. Similarly, the gain of each mmWave path follows a normal distribution with a mean value of 1 and a variance of 1, as $N(0, 1)$. It is assumed that the system operates under a carrier frequency of 28 GHz with a bandwidth of 100 MHz. The simulations consider the massive millimeter wave MIMO system model with the parameters shown in Tab. 1.

The proposed SMP-DF CS algorithm is compared with the existing conventional direct and CoSaMP-based differential CS (CS-DF) algorithms at two different compression ratios. In the differential CIR, it is assumed that the initial amplitude and its support vectors may vary from the value 0 to N and that such vectors are always independent. To maintain the appropriate effective channel compression ratio ($\eta = \frac{\mu}{L}$) over the feedback slots, η is maintained at 25% and 45%, respectively. The SMP recovery CS-based differential feedback scheme is compared with a direct CS scheme and a CS recovery-based differential feedback method. For evaluation, the normalized mean square error (NMSE) is calculated using the following equation.

$$\text{NMSE} = E\left[\frac{||\hat{H} - H||^2}{||H||^2}\right], \tag{19}$$

where $\hat{H}$ is the estimate of the true channel H.

A comparison of NMSE for the specific schemes is shown in Fig. 2 for different signal-to-noise ratio (SNR) regimes.

To achieve effective compression ratios, 45% and 15% are considered for initial and subsequent time slots for the first case (η =25%). For the second case (η =45%), the ratios are 65% and 35%, respectively. This result shows that the proposed scheme outperforms other existing schemes by 36% and 24% at the compression ratios of η at 25% and 45%, respectively. This is achieved by leveraging the structural sparsity of signals as a subspace combination. The adaptive pursuit strategies to select the relevant subspace and coefficient iteratively, integrating the differential operation, enhance overall robustness and improve the recovered accuracy.

The achievable sum rate of the proposed system is compared with all other existing direct CS and differential feedback-based CS schemes at different compression ratios of 25% and 45%, respectively. The achievable sum rate is to maximize the data rate supported by the system for all end users given the channel. Figure 3 shows that as SNR increases, with the achievable sum rate also improving for the proposed scheme. The SMP-based differential CS scheme performs the best across the entire SNR range compared to other existing CS

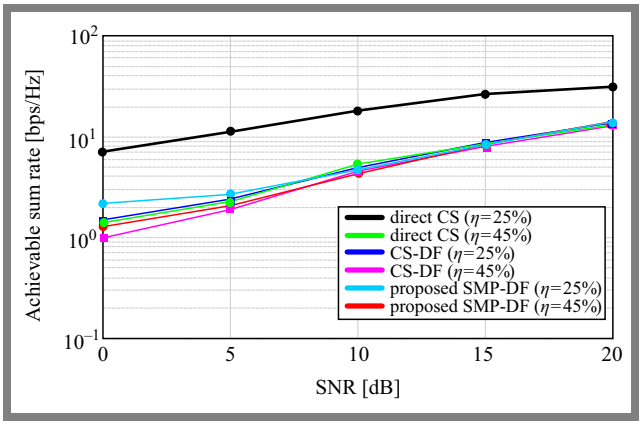

Fig. 3. Achievable sum rate versus SNR of the proposed scheme.

schemes. Furthermore, the gap between the direct CS method and the proposed SMP-based differential feedback method becomes comparatively narrower as SNR increases and the compression ratio η increases from 25% to 45%, respectively.

5. Conclusion and Future Work

This article investigates the sparse channel recovery problem and the issue of channel feedback overhead. To enhance the performance of the CS-based differential channel feedback scheme, the article proposes a subspace matching pursuit recovery algorithm used in conjunction with differential operations. A simulation has been performed to analyze and compare NMSE across different SNR regimes.

Results of the simulation show that the proposed scheme outperforms the other existing direct CS and differential CS schemes, reducing the channel overhead by 36% and 24% at different compression ratios over the time slots of mmWave massive MIMO systems. In future work, the proposed scheme may be extended to intelligent reflecting surface (IRS)-aided mmWave massive MIMO systems to meet the requirements of future-generation wireless systems.

References

[1] *mmWave Massive MIMO: A Paradigm for 5G*, S. Mumtaz, J. Rodriguez, and L. Dai (eds.), Elsevier, 351 p., 2017 (https://doi.org/10.1016/C2015-0-01250-3).

[2] R.W. Heath *et al.*, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems", *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, pp. 436–453, 2016 (https://doi.org/10.1109/JSTSP.2016.2523924).

[3] S.A. Busari *et al.*, "Millimeter-wave Massive MIMO Communication for Future Wireless Systems: A Survey", *IEEE Communications Surveys & Tutorials*, vol. 20, pp. 836–869, 2017 (https://doi.org/10.1109/COMST.2017.2787460).

[4] Y. Shi, M. Badi, D. Rajan, and J. Camp, "Channel Reciprocity Analysis and Feedback Mechanism Design for Mobile Beamforming Systems", *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 6029–6043, 2021 (https://doi.org/10.1109/TVT.2021.3079837).

[5] J. Flordelis *et al.*, "Massive MIMO Performance – TDD versus FDD: What Do Measurements Say?", *IEEE Transactions on Wireless Communications*, vol. 17, pp. 2247–2261, 2018 (https://doi.org/10.1109/TWC.2018.2790912).

[6] H. Xie *et al.*, "Channel Estimation for TDD/FDD Massive MIMO Systems with Channel Covariance Computing", *IEEE Transactions on Wireless Communications*, vol. 17, pp. 4206–4218, 2018 (https://doi.org/10.1109/TWC.2018.2821667).

[7] T. Kim and D.J. Love, "Virtual AoA and AoD Estimation for Sparse Millimeter Wave MIMO Channels", *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Stockholm, Sweden, 2015 (https://doi.org/10.1109/SPAWC.2015.7227017).

[8] X. Cheng, M. Wang, and S. Li, "Compressive Sensing-based Beamforming for Millimeter-wave OFDM Systems", *IEEE Transactions on Communications*, vol. 65, pp. 371–386, 2016 (https://doi.org/10.1109/TCOMM.2016.2616390).

[9] W. Shen *et al.*, "Compressive Sensing-based Differential Channel Feedback for Massive MIMO", *Electronics Letters*, vol. 51, pp. 1824–1826, 2015 (https://doi.org/10.1049/el.2015.0488).

[10] Z. Gao *et al.*, "Compressive Sensing Techniques for Next-generation Wireless Communications", *IEEE Wireless Communications*, vol. 25, pp. 144–153, 2018 (https://doi.org/10.1109/MWC.2017.1700147).

[11] A. Alkhateeb, O. El Ayach, G. Leus, and R.W. Heath, "Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems", *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, pp. 831–846, 2014 (https:/doi.org/10.1109/JSTSP.2014.2334278).

[12] R. Zhang, H. Zhang, W. Xu, and C. Zhao, "A Codebook Based Simultaneous Beam Training for mmWave Multi-user MIMO Systems with Split Structures", *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, UAE, 2018 (https://doi.org/10.1109/GLOCOM.2018.8648139).

[13] J. Wang, S. Kwon, and B. Shim, "Generalized Orthogonal Matching Pursuit", *IEEE Transactions on Signal Processing*, vol. 60, pp. 6202–6216, 2012 (https://doi.org/10.1109/TSP.2012.2218810).

[14] J. Wang and B. Shim, "On the Recovery Limit of Sparse Signals Using Orthogonal Matching Pursuit", *IEEE Transactions on Signal Processing*, vol. 60, pp. 4973–4976, 2012 (https://doi.org/10.1109/TSP.2012.2203124).

[15] I. Kim and J. Choi, "Channel Estimation via Gradient Pursuit for mmWave Massive MIMO Systems with One-bit ADCs", *EURASIP Journal on Wireless Communications and Networking*, art. no. 289, 2019 (https://doi.org/10.1186/s13638-019-1623-x).

[16] S. Sun and T.S. Rappaport, "Millimeter Wave MIMO Channel Estimation Based on Adaptive Compressed Sensing", *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, Paris, France, 2017 (https://doi.org/10.1109/ICCW.2017.7962632).

[17] A. Manoj and A.P. Kannu, "Multi-user Millimeter Wave Channel Estimation Using Generalized Block OMP Algorithm", *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, Japan, 2017 (https://doi.org/10.1109/SPAWC.2017.8227670).

[18] Z. Gao, C. Hu, L. Dai, and Z. Wang, "Channel Estimation for Millimeter-wave Massive MIMO with Hybrid Precoding over Frequency-selective Fading Channels", *IEEE Communications Letters*, vol. 20, pp. 1259–1262, 2016 (https://doi.org/10.1109/LCOMM.2016.2555299).

[19] J.P. González-Coma *et al.*, "Channel Estimation and Hybrid Precoding for Frequency Selective Multiuser mmWave MIMO Systems", *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 353–367, 2018 (https://doi.org/10.1109/JSTSP.2018.2819130).

[20] J. Rodríguez-Fernández, N. González-Prelcic, K. Venugopal, and R.W. Heath, "Frequency-domain Compressive Channel Estimation for Frequency-selective Hybrid Millimeter Wave MIMO Systems", *IEEE Transactions on Wireless Communications*, vol. 17, pp. 2946–2960, 2018 (https://doi.org/10.1109/TWC.2018.2804943).

[21] K. Venugopal, A. Alkhateeb, N. González Prelcic, and R.W. Heath, "Channel Estimation for Hybrid Architecture-based Wideband Millimeter Wave Systems", *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 1996–2009, 2017 (https://doi.org/10.1109/JSAC.2017.2720856).

[22] H. Kim, G.-T. Gil, and Y.H. Lee, "Two-step Approach to Time-domain Channel Estimation for Wideband Millimeter Wave Systems with Hybrid Architecture", *IEEE Transactions on Communications*, vol. 67, pp. 5139–5152, 2019 (https://doi.org/10.1109/TCOMM.2019.2906873).

[23] P. Uthansakul and A.A. Khan, "On the Energy Efficiency of Millimeter Wave Massive MIMO Based on Hybrid Architecture", *Energies*, vol. 12, art. no. 2227, 2019 (https://doi.org/10.3390/en12112227).

[24] X. Song, T. Kühne, and G. Caire, "Fully-/partially-connected Hybrid Beamforming Architectures for mmWave MU-MIMO", *IEEE Transactions on Wireless Communications*, vol. 19, pp. 1754–1769, 2020 (https://doi.org/10.1109/TWC.2019.2957227).

[25] W. Tong *et al.*, "Deep Learning Compressed Sensing-based Beamspace Channel Estimation in mmWave Massive MIMO Systems", *IEEE Wireless Communications Letters*, vol. 11, pp. 1935–1939, 2022 (https://doi.org/10.1109/LWC.2022.3188530).

[26] Z. Zhou *et al.*, "Low-rank Tensor Decomposition-aided Channel Estimation for Millimeter Wave MIMO-OFDM Systems", *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 1524–1538, 2017 (https://doi.org/10.1109/JSAC.2017.2699338).

[27] F. Gomez-Cuba and A.J. Goldsmith, "Sparse mmWave OFDM Channel Estimation Using Compressed Sensing", *ICC 2019–2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019 (https://doi.org/10.1109/ICC.2019.8761440).

[28] D. Sacristán-Murga and A. Pascual-Iserte, "Differential Feedback of MIMO Channel Gram Matrices Based on Geodesic Curves", *IEEE Transactions on Wireless Communications*, vol. 9, pp. 3714–3727, 2010 (https://doi.org/10.1109/TWC.2010.102210.091686).

[29] X. Li, J. Fang, H. Li, and P. Wang, "Millimeter Wave Channel Estimation via Exploiting Joint Sparse and Low-rank Structures", *IEEE Transactions on Wireless Communications*, vol. 17, pp. 1123–1133, 2017 (https://doi.org/10.1109/TWC.2017.2776108).

[30] T. Jiang, M. Song, X. Zhao, and X. Liu, "Channel Estimation for Millimeter Wave Massive MIMO Systems Using Separable Compressive Sensing", *IEEE Access*, vol. 9, pp. 49738–49749, 2021 (https://doi.org/10.1109/ACCESS.2021.3069335).

[31] V. Baranidharan *et al.*, "Modified Compressive Sensing and Differential Operation Based Channel Feedback Scheme for Massive MIMO Systems for 5G Applications", *Proc. of the Fourth International Conference on Smart Computing and Informatics*, pp. 251–260, 2021 (https://doi.org/10.1007/978-981-16-0878-0_25).

[32] Y. Han, W. Shin, and J. Lee, "Projection-based Differential Feedback for FDD Massive MIMO Systems", *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 202–212, 2016 (https://doi.org/10.1109/TVT.2016.2542195).

[33] L. Zhang, L. Song, M. Ma, and B. Jiao, "On the Minimum Differential Feedback for Time-correlated MIMO Rayleigh Block-fading Channels", *IEEE Transactions on Communications*, vol. 60, pp. 411–420, 2012 (https://doi.org/10.1109/TCOMM.2012.011311.100455).

[34] W. Ma, C. Qi, and G.Y. Li, "High-resolution Channel Estimation for Frequency-selective mmWave Massive MIMO Systems", *IEEE Transactions on Wireless Communications*, vol. 19, pp. 3517–3529, 2020 (https://doi.org/10.1109/TWC.2020.2974728).

[35] J. Mo, P. Schniter, and R.W. Heath, "Channel Estimation in Broadband Millimeter Wave MIMO Systems with Few-bit ADCs", *IEEE Transactions on Signal Processing*, vol. 66, pp. 1141–1154, 2017 (https://doi.org/10.1109/TSP.2017.2781644).

[36] H. Xie and N. González-Prelcic, "Dictionary Learning for Channel Estimation in Hybrid Frequency-selective mmWave MIMO Systems", *IEEE Transactions on Wireless Communications*, vol. 19, pp. 7407–7422, 2020 (https://doi.org/10.1109/TWC.2020.3011126).

[37] B. Qi, W. Wang, and B. Wang, "Off-grid Compressive Channel Estimation for mm-Wave Massive MIMO with Hybrid Precoding", *IEEE Communications Letters*, vol. 23, pp. 108–111, 2018 (https://doi.org/10.1109/LCOMM.2018.2878557).

[38] K. Venugopal, A. Alkhateeb, R.W. Heath, and N. González Prelcic, "Time-domain Channel Estimation for Wideband Millimeter Wave Systems with Hybrid Architecture", *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017 (https://doi.org/10.1109/ICASSP.2017.7953407).

Baranidharan V, M.Tech.
Department of Electronics and Communication Engineering
https://orcid.org/0000-0003-3521-823X
E-mail: svbaranidhar@gmail.com
National Institute of Technology Puducherry, Karaikal, India
https://www.nitpy.ac.in
Bannari Amman Institute of Technology, Sathy, India
https://www.bitsathy.ac.in

Surendar M, Ph.D.
Department of Electronics and Communication Engineering
https://orcid.org/0000-0001-5561-7516
E-mail: surendar.m@nitpy.ac.in
National Institute of Technology Puducherry, Karaikal, India
https://www.nitpy.ac.in

Optimizing Spectral and Energy Efficiency of Massive MIMO Networks Using MVO and API Algorithms

Hiba Ines Bitat[1], Fouzia Maamri[1], Fatima Khelfaoui[2], Hanane Djellab[3], Yacine Belhocine[4], and Yacine Messai[1]

[1]*SATIT Laboratory, University of Abbes Laghrour, Khenchela, Algeria,*
[2]*LTPh Laboratory, University of Abbes Laghrour, Khenchela, Algeria,*
[3]*LTI Laboratory, University of Larbi Tebessi, Tebessa, Algeria,*
[4]*LAMIS Laboratory, University of Larbi Tebessi, Tebessa, Algeria*

https://doi.org/10.26636/jtit.2025.1.1993

Abstract — Wireless communication, especially that relying on 5G technology, plays a crucial role in modern networks. The use of massive multiple-input, multiple-output (MIMO) systems is one of the key advancements in this area, as it improves energy efficiency (EE) and spectral efficiency (SE), making such a technique critical for future communication networks. This article focuses on optimizing EE and SE using a new metaheuristic multiverse optimization algorithm (MVO), and compares the results obtained with those achieved with the use of the Pachycondyla Apicalis algorithm (API) and other methods. Furthermore, the study explores the best values for factors such as coherence time, power amplifier efficiency, and hardware power in each user, with all of them playing a critical role in maximizing EE. The authors also examine the correlation between EE and SE in the downlink direction. The results show that the MVO approach achieves better performance in fewer iterations compared to API and other methods, demonstrating its potential for improving wireless communication systems.

Keywords — 5G, energy efficiency, massive MIMO, multiverse optimizer, Pachycondyla Apicalis algorithm, spectral efficiency

1. Introduction

In recent years, the number of electronic devices connected to the Internet has grown quite rapidly. The fact that mobile phones, machines, cars, drones, and many other devices are connected to the web creates several challenges. We face such issues as higher amounts of interference, poor power efficiency, high propagation losses, and low communication efficiency [1], [2]. Traditional antennas are not capable of handling this massive increase in the number of devices they serve. Therefore, it is crucial to improve antennas and adopt new technologies that can manage such large numbers of connections.

Technologies such as massive multiple input multiple output (MIMO), beamforming, and precoding are key solutions [3]. These advances are part of new radio (NR) systems which are designed to support the growing number of users. They are capable of providing high data rates and improving spectral efficiency by at least ten times [4].

This paper focuses on using the massive MIMO technology to address the challenges faced. In a communication system relying on massive MIMO, the base station (BS) and the users interact in a way that utilizes many antennas in the BS to improve signal quality and efficiency [5]. When a user sends a request or data, the base station uses its large number of antennas to transmit the data to the user. These antennas work together to simultaneously send multiple signals to different users or even the same user, but using different channels or frequencies [6], [7]. This process is called "beamforming", where the BS can direct its signal, in a focused manner, to a specific user, thus improving the strength of the signal and reducing interference from other users [8].

On the user's side, a device like a smartphone has its own antennas, and when it receives the BS signal, it decodes the data. Communication occurs in such a way that the BS can serve many users simultaneously, each with a dedicated and stronger signal. Such a mode of operation improves network efficiency, ensures faster data speeds and offers more reliable connections [9].

The trade-off between spectral efficiency (SE) and energy efficiency (EE) is an important factor we focus on in this study. An increase in SE affects EE. In this paper, we aim to find the optimal balance between SE and EE. Specifically, we want to identify the ideal combination between the number of users and antennas and the transmission power to achieve the best values of both SE and EE. High SE and high EE are critical for the success of 5G networks, as they ensure faster data rates and lower energy consumption, improving overall network performance.

The proposed approach uses two different metaheuristic algorithms, i.e. multiverse optimization and Apicalis Pachycondyla, to enhance both SE and EE. We analyze how different parameters, such as power amplifier efficiency, coherence time, and hardware power at each user, affect EE. By examining the relationship between SE and EE, we aim to find the critical point that gives the highest values for both factors. Finally, the EE-SE trade-off is derived in a closed form, re-

ducing computational complexity by expressing the essential derivatives in terms of power, thus making it easier to find the optimal solution.

This paper is organized as follows. Section 2 reviews related articles, summarizes their main ideas, and describes the methods used. Section 3 provides an overview of the key aspects of massive MIMO. In Section 4, we explain the functioning of two metaheuristic algorithms: the multiverse optimizer (MVO) and Pachycondyla APIcalis (API). Section 5 presents the simulation results and compares the performance of MVO and API with other methods. Finally, Section 6 concludes the article.

2. Related Works

Numerous studies have been dedicated to improving energy efficiency in massive MIMO systems. For example, article [10] explores hybrid systems that combine massive MIMO with other technologies to improve EE. Similarly, paper [11] proposed two energy efficient beamforming algorithms for multi-user downlink systems, aiming to improve EE while meeting SINR constraints. The methods investigated showed better results than traditional beamforming techniques and pointed out the need to study the effect of circuit power further. In addition, in articles [12], [13], the authors did not look at spectral efficiency (SE).

On the other hand, some research focused solely on improving SE. For example, [14] studied how massive multiuser MIMO systems perform during uplink transmission, when the base station is equipped with many antennas and each user has just one. They created methods to improve SE. In addition, in [15], the goal is to improve the number of users that may connect and communicate efficiently within a given network. The researcher proposes a new design that groups users by location or similar characteristics and serves each group with the best-suited method. This approach reduces the required number of antennas, while still achieving high efficiency and providing better performance compared to older methods [16]. This article aims to reduce power usage in massive MIMO systems by using low-resolution (2-bit) ADCs. It studies how these ADCs affect spectral efficiency of the system's uplink under different conditions, such as perfect and imperfect knowledge of the communication channel. The proposed method involves mathematical modeling and formulas to predict SE performance, showing that even with low-cost ADCs, good results can be achieved in massive MIMO systems. All this work achieves good results, but does not take into account.

Some studies have explored the balance between spectral efficiency (SE) and energy efficiency (EE) in massive MIMO systems. For example, in [17], the researchers used deterministic and analytical methods focusing on power allocation and selection of access points (AP) through closed form derivations and system constraints. However, they did not use metaheuristic algorithms, relying instead on structured optimization techniques to enhance energy efficiency in cell-free massive MIMO systems. Similarly, in [4], the authors addressed the challenge of optimizing resource efficiency (RE) in a single-cell massive MIMO downlink transmission. Their work considered statistical channel state information at the transmitter (CSIT) to find a balance between SE and EE using mathematical optimization and algorithmic design.

In study [5], the trade-off between SE and EE is analyzed by solving a multi-objective optimization problem. The paper investigates how transmit power and the number of antennas impact this trade-off by examining their first derivatives. In the same context, article [18] used geometric programming to optimize SE and EE in a unified massive cell-free MIMO system with simultaneous wireless information and power transfer (SWIPT). Like the remaining studies mentioned, the work did not employ metaheuristic algorithms.

Although these studies contributed valuable information about SE and EE optimization, they all relied on structured or mathematical methods rather than metaheuristic algorithms. In contrast, our work introduces two novel metaheuristic algorithms in this field: the multiverse optimization (MVO) algorithm, which has not been applied to massive MIMO systems before, and the Pachycondyla Apicalis (API) algorithm. We compare the performance of these algorithms to determine which one achieves better results.

3. Massive MIMO

Massive MIMO is a fundamental innovation in modern wireless communication systems [19], [20]. It addresses the growing demand for high-speed data and reliable connections, i.e. challenges that traditional single-input single-output (SISO) systems were not capable of overcoming due to their limited data rates and inability to support multiple users simultaneously [21].

To overcome the limitations of SISO, advanced MIMO technologies, such as single-user MIMO (SU-MIMO) [22], multiuser MIMO (MU-MIMO) [23], and network MIMO [24], were developed. These technologies improved capacity but struggled with the exponential growth in wireless users and data demands. With billions of connected devices, including devices connected to the Internet of Things (IoT) in smart homes, healthcare, and energy systems, more efficient solutions have become essential [19].

Massive MIMO extends traditional MIMO by deploying hundreds or even thousands of antennas at the base station [25], [26]. This setup improves wireless performance by better focusing energy into smaller spatial regions, thus enhancing spectral efficiency and throughput. Narrowing and directing beams to target users also reduces interference and improves connectivity.

Massive MIMO operates efficiently using time division duplexing (TDD) which divides communication into three main phases during a coherence interval [27]. In the first phase, called channel estimation, users send unique pilot sequences to the base station (BS). The BS estimates the channel state information (CSI) using these pilots. Accurate CSI enables

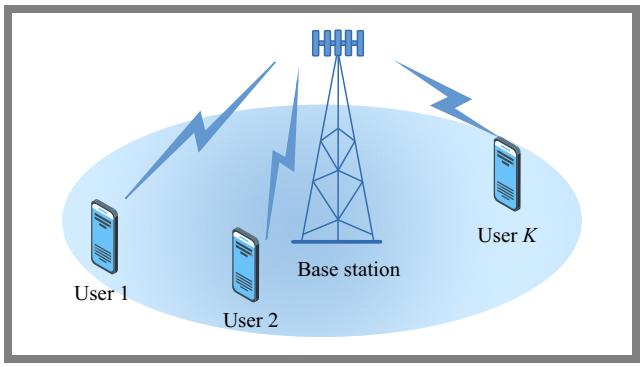

Fig. 1. Massive MIMO transmission concept.

precise signal precoding for the downlink phase. During downlink transmission, the BS uses channel estimates and user-specific data to create pre-coded signals that are transmitted through multiple antennas to all users in the same time-frequency resource. This process improves communication efficiency and reliability by using channel information to reduce interference and optimize energy usage.

Massive MIMO offers several key advantages over traditional MIMO systems. It provides higher spectral efficiency by reusing time-frequency resources for multiple users. It also improves energy efficiency by focusing beams to reduce power wastage and interference. Additionally, it improves reliability by supporting more users with stable connections. These advantages make massive MIMO indispensable for 5G and future networks [19], [20].

Figure 1 illustrates the downlink transmission in a massive MIMO setup. The BS in a cell transmits the downlink signal x_l as follows:

$$x_l = \sum_{k=1}^{K} w_{kl}\, q_k \ , \tag{1}$$

where $q_k \sim CN(0, \rho_{kl})$ represents the data signal intended for user equipment (UE) k in cell l, and $w_{kl} \in C^M$ is the precoding vector directing the signal. The precoding vector satisfies $E[\,|w_{kl}|^2] = 1$, ensuring $E[\,|x_l|^2] = \rho_{kl}$, which corresponds to the transmit power for UE k. The received signal y_{ik} at UE k in cell i is:

$$y_{ik} = h_{iilk}^H\, w_{kl}\, q_k + \sum_{l \neq i} \sum_{j=1}^{K} h_{iilk}^H\, w_{jl}\, q_j + n_{ik} \ , \tag{2}$$

where h_{iilk} denotes the channel between BS l and UE k in cell i, and $n_{ik} \sim CN(0, \sigma^2)$ represents additive noise. The received signal consists of the desired signal, interference between cells, and noise [27].

3.1. Energy Efficiency in Wireless Networks

Energy efficiency (EE) is a key feature in modern wireless networks, especially after the introduction of the 5G technology. EE measures how much data can be transmitted using a certain amount of energy [28], [29]. It is calculated as:

$$EE = \frac{\text{Throughput [bit/s/cell]}}{\text{Power consumption [W/cell]}} \ . \tag{3}$$

This ratio, expressed in bits per Joule, helps reduce costs and environmental impact. Improving EE involves techniques such as setting base stations to sleep mode when traffic is low, using renewable energy, and optimizing resources such as antennas, spectrum, and power.

The MIMO technology is a major contributor to improving EE, as it increases spectral efficiency. This means that more data can be sent using the same amount of energy, making networks more energy efficient [22], [30]–[32].

To model and calculate EE, we use the following objective function [33]:

$$EE = \frac{N * AUR}{P_{tot}} \ , \tag{4}$$

where N is the number of active antennas, AUR is the average user rate, and P_{tot} is the total power consumption. These values are defined using the following specific equations.

The average user rate represents the average data rate for each user, influenced by factors such as the number of antennas and the transmitting power. It is calculated as [33]:

$$AUR = R_{avg} = \omega \left(1 - \frac{N}{\omega_c \cdot t_c}\right) \log_2 \left(1 + \frac{\frac{p_t}{N}(M - N)}{p_n^2 \Psi_1 + p_t \Psi_2}\right) \ . \tag{5}$$

Total power consumption includes all sources of power used in the network, such as transmission, hardware, and processing. It is given by [33]:

$$P_{tot} = \frac{p_t}{\eta} + M p_c^M + N p_c^N + \frac{FP}{\eta_c} + p_s \ . \tag{6}$$

Floating point processes represent the computational load of the system, calculated as [33]:

$$FP = 3\, N^2 M \frac{\omega}{\omega_c\, t_c} \ . \tag{7}$$

Several parameters influence EE and overall system performance:

- M_{max} – maximum number of antennas at the base station, which determines the capacity of the system. More antennas mean better beamforming and spectral efficiency,
- ω – transmission bandwidth, defining the range of frequencies for communication. A wider bandwidth supports higher data rates,
- p_n^2 – average noise power, which affects signal quality and SNR.
- Ψ_1 and Ψ_2 – these represent channel conditions and inter-cell interference. Good channel conditions Ψ_1 and lower interference Ψ_2 lead to higher efficiency.
- η – power amplifier efficiency, which impacts energy consumption during transmission.
- p_s – static hardware power, representing the baseline energy used by such components as cooling systems.
- η_c – computational efficiency, indicating how effectively the system processes tasks.
- ω_c and t_c – coherence bandwidth and time, critical for stable communication and efficient resource allocation.

By integrating advanced technologies, such as massive MIMO and carefully optimizing the above parameters, we can signifi-

cantly improve EE in wireless networks. This not only reduces energy consumption, but also enhances network performance, making future communication systems more sustainable.

3.2. Spectral Efficiency

Spectral efficiency (SE) measures how efficiently a wireless system uses its available frequency spectrum. It is expressed in bits per second per Hertz (bps/Hz). In massive MIMO systems, the use of large-scale antenna arrays greatly improves SE. The mathematical expression for SE is:

$$SE = \log_2(1 + SINR)\,, \tag{8}$$

where $SINR$ is the signal-to-interference plus noise ratio. Adding more antennas increases $SINR$, leading to higher spectral efficiency [29], [34].

To further improve SE in wireless networks, several techniques can be used. An increased in the number of antennas in a massive MIMO systems enhances spatial multiplexing and reduces interference, improving SINR. Optimizing beamforming techniques also focuses the signal more effectively towards intended users, reducing interference, and boosting SE. Advanced modulation and coding schemes allow more bits to be transmitted per Hertz, additionally increasing SE. Furthermore, advanced interference management methods, such as interference cancellation and coordination between base stations, can improve SE, especially in dense networks [35].

In this work, we use the following objective function to model SE:

$$SE = \frac{N * AUR}{\omega}\,. \tag{9}$$

4. Algorithm Description and Methodology

4.1. Multi-Verse Optimizer Algorithm

In this paper, the multiverse optimizer (MVO) is used, inspired by three key concepts: white holes, black holes, and wormholes. These ideas are integrated into the algorithm's key steps and equations.

White holes help in the exploration process. In cosmology, the Big Bang is considered a white hole, and in the multiverse theory, collisions between universes can create white holes, acting as gateways between them. Universes with high inflation rates are more likely to have white holes that transport objects outwards, unlike black holes that pull things in [36], [37].

Black holes have strong gravitational pulls, trapping objects, including light. They are more common in universes with low inflation rates and can receive objects from white holes. This exchange between white holes and black holes allows to transfer variables between universes [38], [39]. The mechanism is outlined as follows:

$$x_i^j = \begin{cases} x_k^j & r_1 < NI(Ui) \\ x_i^j & r_1 \geqslant NI(Ui) \end{cases}\,. \tag{10}$$

Wormholes act as tunnels, allowing objects to move between different parts of a universe or between universes. In MVO, wormholes randomly transport objects between a universe and the best universe found so far. The probability of a wormhole appearing and the distance at which it moves objects are controlled by two factors [37], [39].

Adaptive wormhole existence probability (WEP) represents the likelihood of wormholes appearing within universes during an optimization process.

$$WEP = min + l \times \frac{max - min}{L}\,. \tag{11}$$

Adaptive traveling distance rate (TDR) controls how far the variables can move from the best solution when using wormholes.

$$TDR = 1 - \frac{l^{\frac{1}{p}}}{L^{\frac{1}{p}}}\,, \tag{12}$$

$$x_i^j = \begin{cases} \begin{cases} x_j + TDR \times \left((ub_j - lb_j) \times r_4 + lb_j\right) \\ r_3 < 0.5 \\ x_j - TDR \times \left((ub_j - lb_j) \times r_4 + lb_j\right) \\ r_3 \geqslant 0.5 \end{cases} & r_2 < \text{WEP} \\ x_i^j & r_2 \geqslant \text{WEP} \end{cases} \tag{13}$$

The general steps of the MVO algorithm are the following:

Initialization. The algorithm starts by defining key parameters for the MVO, such as white hole exploration probability (WEP), travel distance rate (TDR), and the number of universes p. These parameters control the exploration and exploitation during optimization. A random initialization of the universes is performed, where each universe represents a potential solution to the problem.

Normalization. Once the universes have been initialized, they are sorted based on their fitness values (a measure of how good the solution is). The inflation rates of all universes are then normalized to create a probabilistic model, where better universes (higher fitness) are more likely to influence others.

Fitness evaluation. For each universe in the population, the fitness function is evaluated. This function quantifies the quality of the solution represented by the universe, guiding the optimization process.

Loop start. The algorithm enters a loop to iterate through all universes. It starts with $i = 1$, representing the first universe, and processes each one sequentially to update its properties.

Update parameters. WEP and TDR parameters are dynamically updated during the iteration. This adjustment balances exploration (searching new areas) and exploitation (refining known good solutions). The blackhole index, representing the best universe, is identified based on the highest fitness value.

Inner loop. A nested loop begins with $j = 1$, representing the first object (variable) within the universe. The algorithm will iterate through all objects in the universe for potential updates.

Generate random value. A random number r is generated between 0 and 1. This random value determines whether an object in the current universe will be replaced based on white hole probabilities.

Check the probability of a white hole. If the random number r is less than WEP, the object is replaced using a roulette wheel selection mechanism, where objects from better universes (white holes) are more likely to be chosen. Otherwise, the object remains unchanged.

Increment object index. The index j is incremented to process the next object within the universe. If all objects have been processed ($j >$ number of objects), the algorithm proceeds to the next step.

Check universe completion. Index i is incremented to process the next universe. If all universes have been processed ($i >$ number of universes), the algorithm moves to check the stopping criteria.

Check stopping criteria. The algorithm evaluates whether the stopping criteria, such as reaching the maximum number of iterations or convergence to an optimal solution, are met. If the criteria are satisfied, the algorithm finishes to operate. Otherwise, it restarts the loop for the next iteration.

End. The algorithm concludes by outputting the best universe, representing the optimal solution to the problem.

4.2. Pachycondyla Apicalis Algorithm

The API algorithm, inspired by the foraging behavior of Pachycondyla Apicalis ants, efficiently balances exploration and exploitation to solve optimization problems. Each ant operates individually, performing local searches around hunting sites and dynamically updating its strategies based on results. These ants collectively contribute to the search process through implicit and explicit cooperation. Implicitly, their independent exploration diversifies the search, while explicit recruitment allows ants to share high-quality solutions, fostering global optimization [32], [42].

The robustness of the approach is enhanced by a heterogeneous population of ants with varying amplitudes of exploration, A_{local} and A_{site}, which improves adaptability to diverse problem landscapes. Periodic nest movement, acting as a dynamic restart mechanism, prevents stagnation in suboptimal solutions and allows ants to refocus their search efforts on the most promising areas.

Additionally, success-based memory prioritizes productive hunting sites while forgetting unproductive ones, ensuring efficiency. Key functions, such as defining the search space, global exploration O_{rand}, local refinement O_{explo}, and nest movement work together to maximize the objective function, integrating local exploitation with global exploration [40], [41].

The main equations governing the algorithm are described below:

- Random initialization (global search): The nest location N is initialized randomly in the search space S using:

$$x_i = b_i + U[0,1] \times (B_i - b_i)\,, \tag{14}$$

where b_i and B_i are the bounds of the i-th dimension in S, and $U[0,1]$ is a uniform random value.

- Local exploration (neighborhood search): Around a hunting site s, ants refine their search using:

$$x_i' = x_i + U[-0.5,\ 0.5] \times A \times (B_i - b_i)\,, \tag{15}$$

where A is the exploration amplitude.

- Global and local exploration parameters:

$$A_{site}(i) = x_i \times 0.01\,, \tag{16}$$

$$A_{local}(i) = \frac{A_{site}(i)}{10}\,. \tag{17}$$

These parameters govern the range of global and local exploration, respectively.

- Relocation of the nest. The nest is moved to the best solution s^* found after T iterations:

$$N = s^*. \tag{18}$$

- Recruitment (cooperation).

Two ants compare their best sites. If $f(site_i) < f(site_j)$, replace $site_j$ with $site_i$.

Building on these principles, the API algorithm follows a structured sequence of steps to achieve optimization, as outlined below [43], [44]:

1) Initialization – place the nest in the search area. Set the number of ants, hunting sites, and exploration range.
2) Hunting site exploration – ants search around their hunting sites. If they find a better result, they update the site. If not, they choose a new site.
3) Recruitment (optional) – compare the results of two ants. The weaker site is replaced with the stronger one.
4) Nest movement – move the nest to the best location found after a set number of attempts. Start the search again near the new nest.
5) Stopping criterion – stop when the maximum number of attempts is reached or a good enough result is found.

5. Numerical Results and Discussion

Building 5G networks is challenging because we need to balance several goals: service for fast-moving users, high network capacity, and efficient power usage. These goals often conflict with each other, so we rely on multi-objective optimization techniques to find the best possible solutions. In our research, we apply two metaheuristic algorithms: multiverse optimizer (MVO) and Pachycondyla Apicalis (API), using Matlab. We tested different numbers of users, antennas, and transmission power levels to identify the ideal setup. This approach helps us design 5G networks that deliver excellent performance:

$$X = \begin{cases} & 1 \leqslant N \leqslant \frac{M}{2} \\ [NMP_t]^T & 2 \leqslant M \leqslant M_{max} \\ & 0 \leqslant p \leqslant MP_t^{max} \end{cases}$$

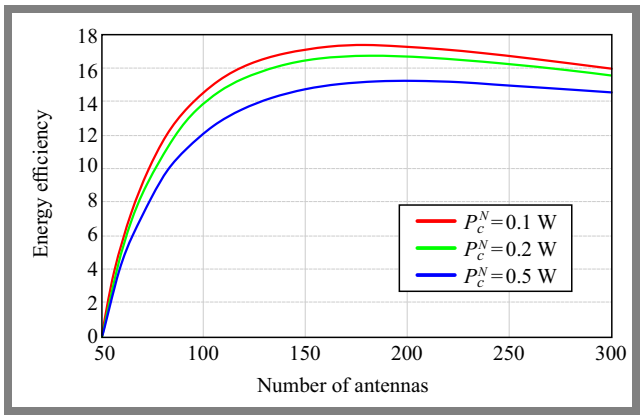

Fig. 2. EE performance versus the number of BS antennas M for different hardware power at each p_c^N .

In this study, we wanted to see how parameter p_c^N affects energy efficiency in a massive MIMO system. We tested three different values: $p_c^N = 0.5$ W, $p_c^N = 0.2$ W, and $p_c^N = 0.1$ W. The value $p_c^N = 0.5$ W represents older hardware, which is less energy efficient. The value $p_c^N = 0.2$ W represents more modern and efficient hardware, while $p_c^N = 0.1$ W represents highly optimized hardware with excellent power efficiency. Our goal was to observe how these values impact energy efficiency as the number of antennas continues to grow.

From the results shown in Fig. 2, we observed that energy efficiency improves significantly as the number of antennas increases, but it eventually reaches a peak and then decreases slightly. Lower values of $p_c^N = 0.1$ W result in higher energy efficiency compared to higher values of 0.5 W. This is the case because more efficient hardware reduces power losses, leading to better overall energy performance. However, after a certain point, adding more antennas starts to increase power consumption without providing significant gains in energy efficiency. In conclusion, using modern, low-power hardware is essential for achieving high energy efficiency in massive MIMO systems, but there is also a limit to how much increasing the number of antennas can improve performance.

In our previous analysis, we studied how p_c^N affects energy efficiency in massive MIMO systems. Now, we want to see how EE is affected by parameter η (Fig. 3). We tested three different values: $\eta = 0.25, 0.35$, and 0.45. The value $\eta = 0.25$ represents older systems, which are less efficient. The value $\eta = 0.45$ represents modern systems with better efficiency. We also introduced an intermediate value, $\eta = 0.35$, to explore a balanced setup. Our goal is to understand how these values influence energy efficiency as the number of antennas increases.

From the results, we observed that energy efficiency improves as the number of antennas increases, similarly to our findings regarding hardware power at each user. However, the efficiency reaches a peak and then starts to decline slightly. Higher values of $\eta = 0.45$ result in better energy efficiency, because they represent more advanced systems with optimized performance. The intermediate value of $\eta = 0.35$ shows ranks between older and modern systems, indicating a balanced trade-off. In conclusion, modern systems with higher η values perform better in terms of energy efficiency, but increasing the number of antennas beyond a certain point does not bring significant improvements and may even slightly reduce efficiency. These findings, combined with our previous analysis of p_c^N, highlight the importance of both hardware efficiency and system parameters in designing energy efficient massive MIMO networks.

After we found the best values of p_c^N and η for energy efficiency in massive MIMO systems, we wanted to see how the coherence time t_C impacts EE – see Fig. 4. We tested three different values: $t_C = 3$ ms, 5 ms, and 7 ms. The value $t_C = 3$ ms represents scenarios with fast-moving users, where the channel conditions change quickly. The value $t_C = 7$ ms represents low mobility scenarios, where the channel remains stable for a longer time. We also tested an intermediate value of $t_C = 3$ ms, to find a balance between these two situations. Our goal is to understand how these coherence time values influence energy efficiency as the number of antennas increases.

From the results, we observed that energy efficiency improves as the number of antennas increases, similarly to our findings concerning p_c^N and η. However, the efficiency eventually reaches a peak and then decreases slightly. Higher values of $t_C = 7$ ms result in better energy efficiency, because more stable channel conditions allow better data transmission with fewer updates. On the other hand, lower values (e.g.,

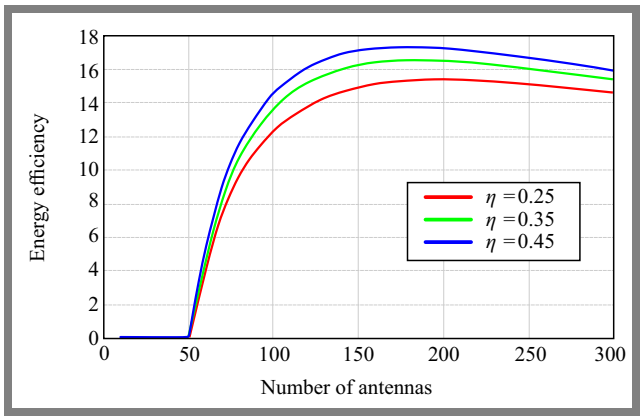

Fig. 3. Impact of power amplifier efficiency η on the corresponding energy efficiency.

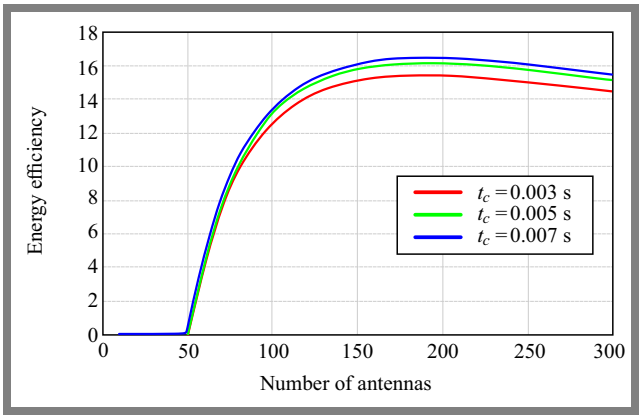

Fig. 4. Energy efficiency as a function of the number of antennas at different coherence times t_C.

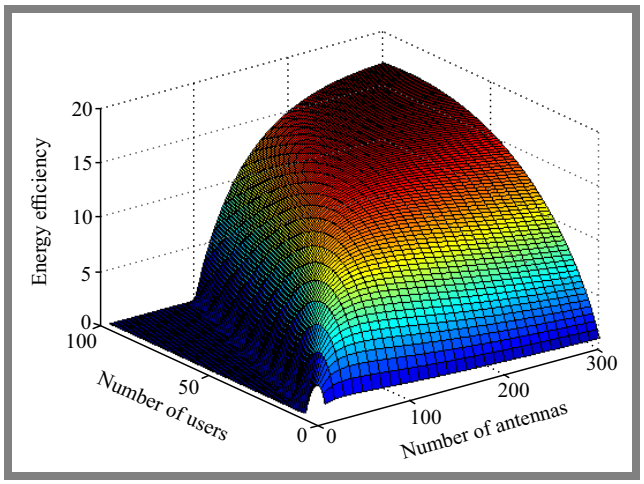

Fig. 5. Energy efficiency as a function of the number of antennas and users in massive MIMO systems.

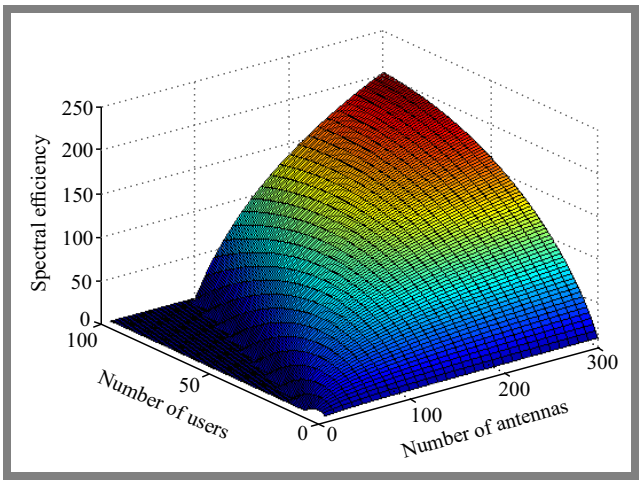

Fig. 6. Spectral efficiency as a function of the number of antennas and users in massive MIMO systems.

3 ms) are characterized by reduced efficiency due to frequent updates and higher overhead. The intermediate value of $t_C = 5$ ms offers a balanced performance ranking between the two extremes.

Table 1 summarizes all the values that we used.

After testing different hardware power per user p_c^N, power amplifier efficiency η, and coherence time t_C values, we selected the best value. Here, we observe how energy efficiency and spectral efficiency change simultaneously as a function of two parameters: number of users N and number of antennas M.

Figure 5 shows how energy efficiency changes with the number of antennas M and the number of users N. When both M and N increase, energy efficiency improves and reaches its highest point. For example, when there are 300 antennas and 100 users, the EE value is 17.01, confirming very good efficiency in a large system. However, with fewer antennas and users, e.g. 20 antennas and 4 users, EE drops to 4.578, showing poor efficiency due to limited resources. In a medium range, with 110 antennas and 37 users, EE reaches 14.49, showing a good balance, but not the highest efficiency. This pattern shows that there is an optimal point where the balance between antennas and users gives the best energy efficiency. Adding more antennas or users after this point does not significantly improve and may even reduce efficiency levels.

Tab. 1. Set of parameters defining the system model.

Parameter	Name	Value
$M_{\max}$	Maximum number of antennas	300
ω	Transmitting bandwidth	10 MHz
p_n^2	Average noise power	10^{-13} W
Ψ_1	Inverse channel loss	$1.72 \cdot 10^9$
Ψ_2	Intercell interference strength	0.540
p_c^N	Hardware power at each user	0.2 W
p_s	Static hardware power	10 W
η_c	Typical computational efficiency	12.8 Gflops/W
ω_C	Bandwidth of coherence	200 kHz
t_C	Coherence time	7 ms
η	The power amplifier's efficiency	0.45
P_C^M	Hardware power consumption	0.5 W

The 3D graph shown in Fig. 6 illustrates how spectral efficiency changes with the number of antennas M and the number of users N.

When both M and N increase, SE improves. For example, at 300 antennas and 100 users, SE reaches 201, showing very good use of the available bandwidth. But with fewer antennas and users, e.g. 20 antennas and 7 users, SE drops to 14.05, showing poor performance due to limited resources.

In the middle range, with 170 antennas and 53 users, SE reaches 117.8, showing a good balance but not the best efficiency. At first, increasing the number of antennas greatly improves SE, but after a certain point, the improvement slows down and efficiency stops growing significantly.

It is important to note that if there are many users and few antennas, the system struggles to manage the users properly, leading to low SE. This happens because a small number of antennas cannot effectively handle many users through beamforming and spatial diversity.

The two graphs presented in Figs. 7 and 8 show how energy efficiency and spectral efficiency are related in a massive MIMO system.

In the 2D graph, as the number of users increases, EE also increases along with SE, but then starts to drop after reaching a peak point. For example, when $N = 60$, the peak occurs at SE = 120 and EE = 16. This happens because adding more users or increasing SE needs more energy and after a point, the system cannot remain efficient.

In the 3D graph, as the number of antennas increases, both SE and EE increase, but only up to a certain limit. After this peak, EE starts to drop, while SE continues to rise. For instance, when $M = 100$, SE = 150 and EE = 12. This shows that achieving a higher SE requires more energy, which limits EE.

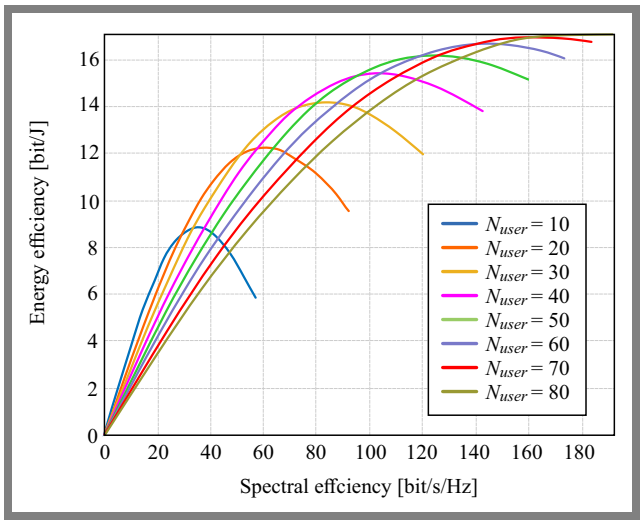

Fig. 7. 2D Energy efficiency vs spectral efficiency for different numbers of users.

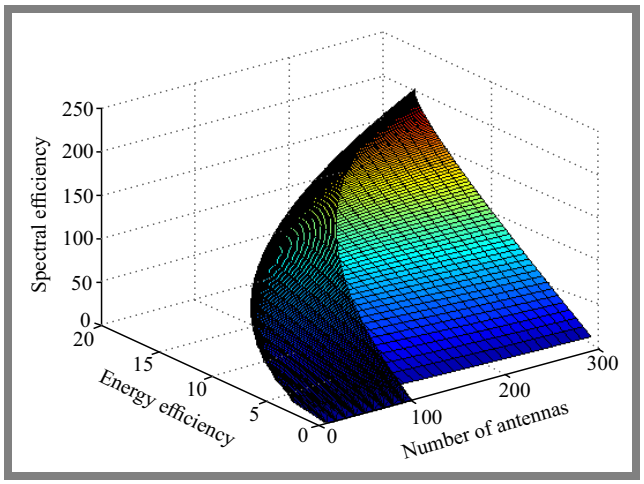

Fig. 8. 3D energy efficiency vs. spectral efficiency for different numbers of users.

This behavior proves that after a certain point, the increase in SE consumes too much power and reduces overall EE. To get the best results, it is important to find the right balance between the number of users and the number of antennas to avoid wasting energy.

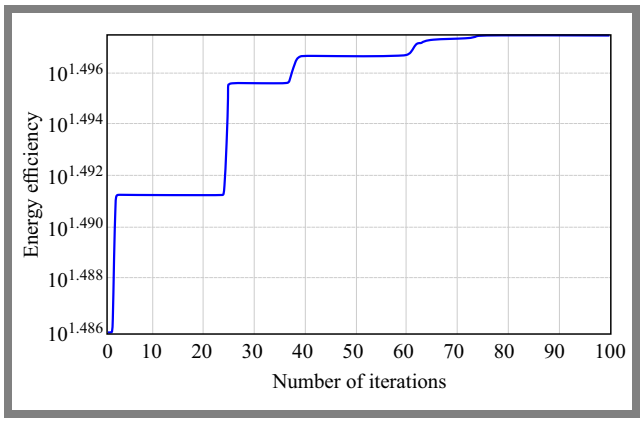

Fig. 9. Optimization of energy efficiency as a function of the number of iterations using the MVO algorithm.

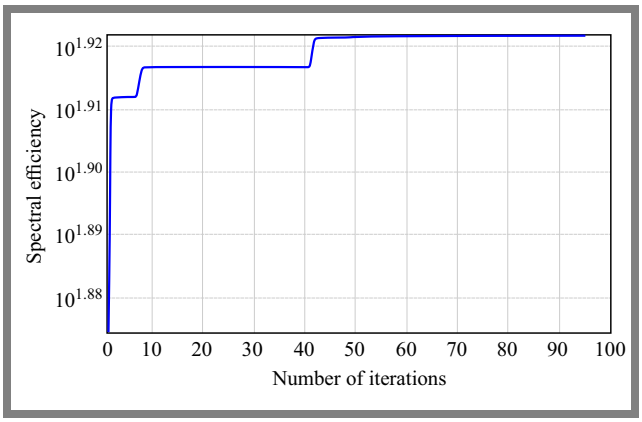

Fig. 10. Optimization of spectral efficiency as a function of the number of iterations using the MVO algorithm.

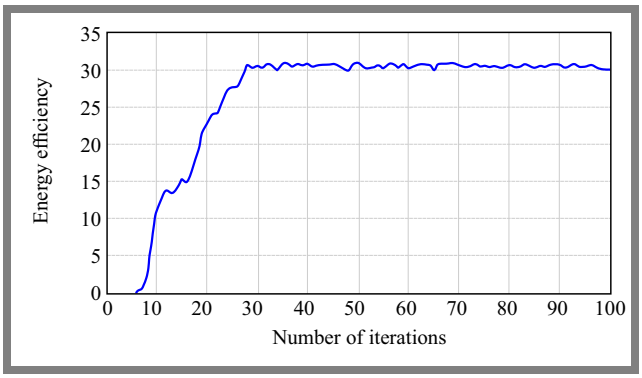

Fig. 11. Optimization of energy efficiency as a function of the number of iterations using the API algorithm.

A comparative analysis of two metaheuristic algorithms, i.e. MVO and API, was performed to evaluate spectral efficiency and energy efficiency. Both algorithms were tested under identical parameters, including the number of iterations set to 100, to ensure a fair comparison (Figs. 9–12). From the results, it is evident that the MVO algorithm outperforms the API in terms of both objectives. Ater 100 iterations, MVO achieved optimal values of SE = 83.55 and EE = 31.44, while API reached SE = 80 and EE = 30. This demonstrates the superior capability of MVO to optimize both spectral and energy efficiency in this scenario.

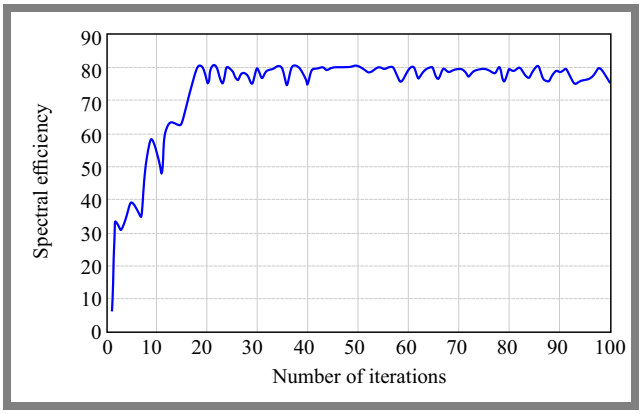

Fig. 12. Optimization of spectral efficiency as a function of the number of iterations using the API algorithm.

Tab. 2. Comparison between MVO and API algorithms for SE.

Criterion	MVO	API
Convergence speed	Faster	Slower
Stability	Stable after 50 iterations	Fluctuates significantly
Final value	∼ 83.55	∼ 78–80
Trend	Smooth growth	Irregular growth
Overall performance	Better	Moderate

Tab. 3. Comparison between MVO and API for EE.

Criterion	MVO	API
Convergence speed	Faster, distinct steps	Gradual, smooth rise
Stability	Perfectly stable after 70 iterations	Minor fluctuations persist
Final value	∼ 31.44	∼ 30–32
Trend	Step-like growth	Smooth growth
Overall performance	Slightly better	Good but less stable

5.1. Comparison of MVO and API

Tables 2–3 compare the performance of MVO and API in terms of spectral efficiency and energy efficiency. As far as SE is concerned, MVO is faster and more stable, achieving a higher final value of approx. 83.55. API is slower, with more fluctuations, and reaches a lower final value of 78 to 80. The growth trend for MVO is smooth, while API shows irregular growth. Overall, MVO performs better in terms of SE.

As far as EE is concerned, MVO also shows better results. It is faster with distinct steps and becomes perfectly stable after 70 iterations. API rises smoothly, but shows minor fluctuations. The final value of EE for MVO is approximately 31.44, which is slightly higher than the API range of 30 to 32. The growth trend for MVO is step-like, while API shows smooth growth. Overall, MVO is better, but API still remains good.

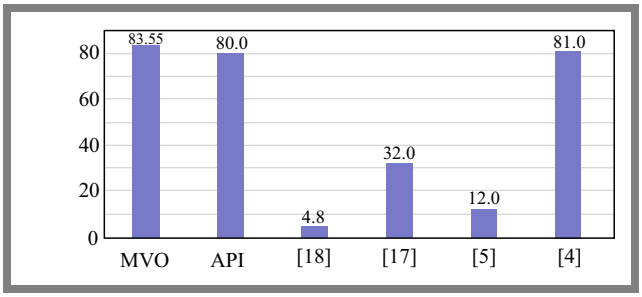

Fig. 13. Comparison of the spectral efficiency values (API and MVO) for the proposed algorithm and other papers.

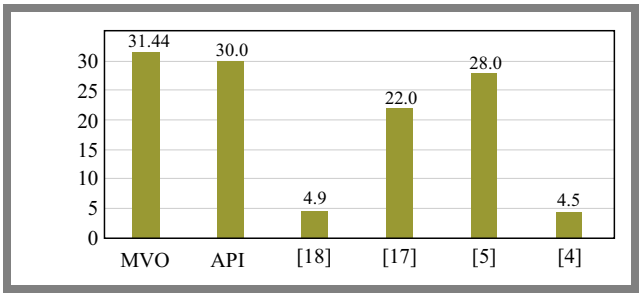

Fig. 14. Comparison of energy efficiency values (API and MVO) for the proposed algorithm and other papers.

6. Conclusions

In this work, we focused on optimizing spectral efficiency and energy efficiency using the massive MIMO technology in the downlink direction. By testing various hardware power, power amplifier efficiency, and coherence time values, we identified the best parameters to achieve an effective balance between SE and EE. Our results showed a trade-off between these two metrics, highlighting their interdependence.

We applied two metaheuristic algorithms, MVO and API, to analyze performance differences. Although both algorithms showed promising results, MVO consistently outperformed API by achieving higher SE and EE values in a shorter lead time. Finally, we compared our findings with previous works, further validating the superior efficiency and stability of the MVO algorithm (Figs. 13–14). This study confirms that MVO is a powerful tool for optimizing massive MIMO systems, offering significant improvements that may benefit modern wireless networks.

Acknowledgments

The authors express their sincere gratitude to the Laboratory of SATIT, Department of Electrical Engineering, Abbes Laghrour University, for providing the necessary resources, facilities, and support in connection with this work. The valuable guidance and assistance received from the laboratory staff and researchers was instrumental in completing this research project.

References

[1] O. Tervo, L.-N. Tran, and M. Juntti, “Optimal Energy-efficient Transmit Beamforming for Multi-user MISO Downlink”, *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5574–5588, 2015 (https://doi.org/10.1109/TSP.2015.2453134).

[2] Q.-D. Vu, L.-N. Tran, R. Farrell, and E.-K. Hong, “Energy-efficient Zero-forcing Precoding Design for Small-cell Networks”, *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 790–804, 2016 (https://doi.org/10.1109/TCOMM.2015.2502941).

[3] M. Mohammadi, Z. Mobini, H.Q. Ngo, and M. Matthaiou, “Ten Years of Research Advances in Full-Duplex Massive MIMO”, *IEEE Transactions on Communications*, early access, 2024 (https://doi.org/10.1109/TCOMM.2024.3464414).

[4] L. You *et al.*, “Spectral Efficiency and Energy Efficiency Tradeoff in Massive MIMO Downlink Transmission with Statistical CSIT”,

IEEE Transactions on Signal Processing, vol. 68, pp. 2645–2659, 2020 (https://doi.org/10.1109/TSP.2020.2986391).

[5] A. Salh *et al.*, "Trade-off Energy and Spectral Efficiency in 5G Massive MIMO System", *arXiv*, 2021 (https://doi.org/10.48550/arXiv.2105.10722).

[6] H. Ju *et al.*, "Transformer-assisted Parametric CSI Feedback for mmWave Massive MIMO Systems", *IEEE Transactions on Wireless Communications*, vol. 23, no. 12, pp. 18774–18787, 2024 (https://doi.org/10.1109/TWC.2024.3476474).

[7] A. Zappone *et al.*, "Energy-efficient Power Control: A Look at 5G Wireless Technologies", *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1668–1683, 2016 (https://doi.org/10.1109/TSP.2015.2500200).

[8] M.A. Saeed and A.O. Nwajana, "A Review of Beamforming Microstrip Patch Antenna Array for Future 5G/6G Networks", *Frontiers in Mechanical Engineering*, vol. 9, 2024 (https://doi.org/10.3389/fmech.2023.1288171).

[9] Z. Behdad, Ö.T. Demir, K.W. Sung, and C. Cavdar, "Interplay Between Sensing and Communication in Cell-Free Massive MIMO with URLLC Users", *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, Dubai, UAE, 2024 (https://doi.org/10.1109/WCNC57260.2024.10571226).

[10] K.N.R.S.V. Prasad, E. Hossain, and V.K. Bhargava, "Energy Efficiency in Massive MIMO-based 5G Networks: Opportunities and Challenges", *IEEE Wireless Communications*, vol. 24, no. 3, pp. 86–94, 2017 (https://doi.org/10.1109/MWC.2016.1500374WC).

[11] C. Jiang and L.J. Cimini, "Downlink Energy-efficient Multiuser Beamforming with Individual SINR Constraints", *MILCOM 2011 – Military Communications Conference*, Baltimore, USA, 2011 (https://doi.org/10.1109/MILCOM.2011.6127719).

[12] E. Björnson, L. Sanguinetti, J. Hoydis, and M. Debbah, "Designing Multi-user MIMO for Energy Efficiency: When is Massive MIMO the Answer?", *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, Istanbul, Türkiye, 2014 (https://doi.org/10.1109/WCNC.2014.6951974).

[13] D. Ha, K. Lee, and J. Kang, "Energy Efficiency Analysis with Circuit Power Consumption in Massive MIMO Systems", *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, London, UK, 2013 (https://doi.org/10.1109/PIMRC.2013.6666272).

[14] O. Saatlou, "Spectral Efficiency Maximization of a Massive Multiuser MIMO System via Appropriate Power Allocation", *Ph.D. thesis, Concordia University*, 2019 (https://spectrum.library.concordia.ca/985700/).

[15] H. Huh, G. Caire, H.C. Papadopoulos, and S.A. Ramprashad, "Achieving 'Massive MIMO' Spectral Efficiency with a Not-so-large Number of Antennas", *IEEE Transactions on Wireless Communications*, vol. 11, no. 9, pp. 3226–3239, 2012 (https://doi.org/10.1109/TWC.2012.070912.111383).

[16] J. Zhang, L. Dai, S. Sun, and Z. Wang, "On the Spectral Efficiency of Massive MIMO Systems with Low-resolution ADCs", *IEEE Communications Letters*, vol. 20, no. 5, pp. 842–845, 2016 (https://doi.org/10.1109/LCOMM.2016.2535132).

[17] H.Q. Ngo *et al.*, "On the Total Energy Efficiency of Cell-free Massive MIMO", *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 25–39, 2018 (https://doi.org/10.1109/TGCN.2017.2770215).

[18] N. Li, Y. Gao, and K. Xu, "On the Optimal Energy Efficiency and Spectral Efficiency Trade-off of CF Massive MIMO SWIPT System", *EURASIP J. on Wireless Communications and Networking*, art. no. 167, 2021 (https://doi.org/10.1186/s13638-021-02035-w).

[19] E.G. Larsson, O. Edfors, F. Tufvesson, and T.L. Marzetta, "Massive MIMO for Next Generation Wireless Systems", *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014 (https://doi.org/10.1109/MCOM.2014.6736761).

[20] N.H.M. Adnan, I. Md. Rafiqul, and A.H.M.Z. Alam, "Massive MIMO for Fifth Generation (5G): Opportunities and Challenges", *2016 International Conference on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, Malaysia, 2016 (https://doi.org/10.1109/ICCCE.2016.23).

[21] S. Labed and N. Aounallah, "Efficient Iterative Detection Based on Conjugate Gradient and Successive Over-relaxation Methods for Uplink Massive MIMO Systems", *Journal of Telecommunications and Information Technology*, no. 2, 2023 (https://doi.org/10.26636/jtit.2023.169023).

[22] R. Chataut and R. Akl, "Massive MIMO Systems for 5G and beyond Networks – Overview, Recent Trends, Challenges, and Future Research Direction", *Sensors*, vol. 20, no. 10, art. no. 2753, 2020 (https://doi.org/10.3390/s20102753).

[23] M. Jiang and L. Hanzo, "Multiuser MIMO-OFDM for Next-generation Wireless Systems", *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1430–1469, 2007 (https://doi.org/10.1109/JPROC.2007.898869).

[24] Z. Zhang *et al.*, "Intelligent Omni Surfaces Assisted Integrated Multi-target Sensing and Multi-user MIMO Communications", *IEEE Transactions on Communications*, vol. 72, no. 8, pp. 4591–4606, 2024 (https://doi.org/10.1109/TCOMM.2024.3374351).

[25] E. Björnson, E.G. Larsson, and T.L. Marzetta, "Massive MIMO: Ten Myths and One Critical Question", *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, 2016 (https://doi.org/10.1109/MCOM.2016.7402270).

[26] T. Van Chien and E. Björnson, "Massive MIMO Communications", in: *5G Mobile Communications*, Springer, pp. 77–116, 2017 (https://doi.org/10.1007/978-3-319-34208-5_4).

[27] F.A. Pereira de Figueiredo, "An Overview of Massive MIMO for 5G and 6G", *IEEE Latin America Transactions*, vol. 20, no. 6, pp. 931–940, 2022 (https://doi.org/10.1109/TLA.2022.9757375).

[28] A. Ashraf *et al.*, "Advancements and Challenges in Scalable Modular Antenna Arrays for 5G Massive MIMO Networks", *IEEE Access*, vol. 12, pp. 57895–57916, 2024 (https://doi.org/10.1109/ACCESS.2024.3391945).

[29] A. Shaikh and M.J. Kaur, "Comprehensive Survey of Massive MIMO for 5G Communications", *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai, UAE, 2019 (https://doi.org/10.1109/ICASET.2019.8714426).

[30] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency", *Foundations and Trends in Signal Processing*, vol. 11, no. 3–4, pp. 154–655, 2017 (https://doi.org/10.1561/2000000093).

[31] J. Tang, D.K.C. So, E. Alsusa, and K.A. Hamdi, "Resource Efficiency: A New Paradigm on Energy Efficiency and Spectral Efficiency Tradeoff", *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4656–4669, 2014 (https://doi.org/10.1109/TWC.2014.2316791).

[32] P. Blacher, E. Lecoutey, D. Fresneau, and E. Nowbahari, "Reproductive Hierarchies and Status Discrimination in Orphaned Colonies of Pachycondyla Apicalis Ants", *Animal Behaviour*, vol. 79, pp. 99–105, 2010 (https://doi.org/10.1016/j.anbehav.2009.10.008).

[33] K.E. Purushothaman and V. Nagarajan, "Multiobjective Optimization Based on Self-organizing Particle Swarm Optimization Algorithm for Massive MIMO 5G Wireless Network", *International Journal of Communication Systems*, vol. 34, art. no. 4725, 2021 (https://doi.org/10.1002/dac.4725).

[34] Y. Liu, B. Ai, and J. Zhang, "Downlink Spectral Efficiency of Massive MIMO Systems with Mutual Coupling", *Electronics*, vol. 12, no. 6, art. no. 1364, 2023 (https://doi.org/10.3390/electronics12061364).

[35] J. Tang, D.K.C. So, E. Alsusa, and K.A. Hamdi, "Resource Efficiency: A New Paradigm on Energy Efficiency and Spectral Efficiency Tradeoff", *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4656–4669, 2014 (https://doi.org/10.1109/TWC.2014.2316791).

[36] D.M. Eardley, "Death of White Holes in the Early Universe", *Physical Review Letters*, vol. 33, art. no. 442, 1974 (https://doi.org/10.1103/PhysRevLett.33.442).

[37] P.J. Steinhardt and N. Turok, "A Cyclic Model of the Universe", *Science*, vol. 296, no. 5572, pp. 1436–1439, 2002 (https://doi.org/10.1126/science.1070462).

[38] R.M. Wald, "The Thermodynamics of Black Holes", *Living Reviews in Relativity*, vol. 4, art. no. 6, 2001 (https://doi.org/10.12942/lrr-2001-6).

[39] S. Mirjalili, S.M. Mirjalili, and A. Hatamlou, "Multi-verse Optimizer: A Nature-inspired Algorithm for Global Optimization", *Neural Computing and Applications*, vol. 27, pp. 495–513, 2016 (https://doi.org/10.1007/s00521-015-1870-7).

[40] N. Monmarché, G. Venturini, and M. Slimane, "On how Pachycondyla Apicalis Ants Suggest a New Search Algorithm", *Future Generation Computer Systems*, vol. 16, no. 8, pp. 937–946, 2000 (https://doi.org/10.1016/S0167-739X(00)00047-9).

[41] F. Maamri, S. Bououden, M. Chadli, and I. Boulkaibet, "The Pachycondyla Apicalis Metaheuristic Algorithm for Parameters Identification of Chaotic Electrical System", *International Journal of Parallel, Emergent and Distributed Systems*, vol. 33, no. 5, pp. 490–502, 2018 (https://doi.org/10.1080/17445760.2017.1401622).

[42] S. Azzeddinne and M. Zoubir, "Pachycondyla APIcalis Ants (API) Algorithm for Multi-user Detection of SDMA-OFDM System", *International Journal of Circuits, Systems and Signal Processing*, vol. 11, pp. 230–235, 2017 (https://www.naun.org/main/NAUN/circuitssystemssignal/2017/a602005-abv.pdf).

[43] H.I. Bitat *et al.*, "Optimization of a MIMO Antenna using the API metaheuristic algorithm", *2024 8th International Conference on Image and Signal Processing and their Applications (ISPA)*, Biskra, Algeria, 2024 (https://doi.org/10.1109/ISPA59904.2024.10536777).

[44] Y. Messai *et al.*, "Use of Meta-heuristic Algorithm to Optimize Ericsson Propagation Model Application on LTE", *2024 8th International Conference on Image and Signal Processing and their Applications (ISPA)*, Biskra, Algeria, 2024 (https://doi.org/10.1109/ISPA59904.2024.10536776).

Hiba Ines Bitat, Ph.D. Student
Laboratory of Systems and Applications of Information and Telecommunication Technologies (SATIT), Department of Electrical Engineering
https://orcid.org/0009-0008-6081-5822
E-mail: bitat.hibaines@univ-khenchela.dz
SATIT Laboratory, University of Abbes Laghrour, Khenchela, Algeria
https://univ-khenchela.com

Fouzia Maamri, Ph.D.
Laboratory of Systems and Applications of Information and Telecommunication Technologies (SATIT), Department of Electrical Engineering
https://orcid.org/0009-0006-8011-4172
E-mail: maamri_fouzia@univ-khenchela.dz
SATIT Laboratory, University of Abbes Laghrour, Khenchela, Algeria
https://univ-khenchela.com

Fatima Khelfaoui, Ph.D.
Phase Transformations Laboratory (LTPh), Department of Electrical Engineering
https://orcid.org/0009-0001-5673-5706
E-mail: khelfaoui.fatima@univ-khenchela.dz
LTPh Laboratory, University of Abbes Laghrour, Khenchela, Algeria
https://www.umc.edu.dz

Hanane Djellab, Ph.D.
Laboratory of LTI Guelma, Department of Electrical Engineering
https://orcid.org/0000-0001-7952-6598
E-mail: hanane.djellab@univ-tebessa.dz
LTI Laboratory, University of Larbi Tebessi, Tebessa, Algeria
https://www.univ-tebessa.dz

Yacine Belhocine, Ph.D. Student
Laboratory of Mathematics Informatics and Systems (LAMIS), Department of Electronics and Telecommunications
https://orcid.org/0009-0002-1747-6008
E-mail: yacine.belhocine@univ-tebessa.dz
LAMIS Laboratory, University of Larbi Tebessi, Tebessa, Algeria
https://univ-khenchela.com

Yacine Messai, Ph.D. Student
Laboratory of Systems and Applications of Information and Telecommunication Technologies (SATIT), Department of Electrical Engineering
https://orcid.org/0009-0005-2808-7560
E-mail: messai.yacine@univ-khenchela.dz
SATIT Laboratory, University of Abbes Laghrour, Khenchela, Algeria
https://univ-khenchela.com

Information for Authors

Journal of Telecommunications and Information Technology (JTIT) is published quarterly since 2000. It comprises original contributions, dealing with a wide range of topics related to telecommunications and information technology. **All papers are subject to peer review.** Topics presented in the JTIT report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

JTIT is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, voice communications devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology.

We encourage submissions from a diverse range of authors from across all countries and backgrounds.

Manuscript

Latex files are preferred and Editorial Office provides a style to prepare the material along with the documentation. We also accept Microsoft Word and PDF files. A typical article is 10 pages long (approximately 6,000 words) and must include the following contents:

- Authors' names and affiliations in the following format:
  First name and surname (last name), academic title,
  Position held,
  ORCID number,
  E-mail address from the University's domain,
  Faculty and name of the University,
  Link to University website.
- Abstract (150-200 words). The abstract should contain statement of the problem, assumptions and methodology, results and conclusion or discussion on the importance of the results. Abstracts must not include mathematical expressions or bibliographic references.
- Keywords related to the content of the article. About four keywords or phrases in alphabetical order should be used, separated by commas.
- The content of the article in a typical structure, i.e.: introduction, related work, conducted research, conclusions, references.

Figures, Tables and Photos

Together with the article, please send files with graphics with the highest resolution available, 150 dpi or more in bitmap resolution (jpg, png) and vector (cdr, svg, ps, pdf) formats are welcomed.

References

We use four main citation styles for a journal article, for an Internet article, for a conference paper, and for a book. Below are examples of citations. In each item, the DOI number or link to the PDF of the cited article should be provided.

[1] R.K. Meyers and A.H. Desoky, "An implementation of the blowfish cryptosystem", *2008 IEEE International Symposium on Signal Processing and Information Technology*, 2008 ((https://doi.org/10.1109/ISSPIT.2008.4775664).

[2] K. Nowicki and T. Uhl, *Ethernet End-to-End*, 1st ed. Germany, Shaker-Publisher, 2008 (ISBN: 978383832271404).

[3] C. Shorten and T.M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019 (https://doi.org/10.1186/s40537-019-0197-0).

[4] S. Wong *et al.*, "Traffic forecasting using vehicle-to-vehicle communication", *3rd Annual Conference on Learning for Dynamics and Control*, pp. 917–929, 2021 (https://arxiv.org/pdf/2104.05528).

Submission

The paper with full PDF version and anonymous PDF version for the blind review process should be submitted on the JTIT website https://www.jtit.pl/jtit/about/submissions.

Reviewing Process

The article is initially approved by the Editor-In-Chief and if the decision is positive, is then sent to the reviewers. Depending on the subject of the article, it takes few weeks. In the next step, reviews are showed to authors who have 2 weeks to correct the article. Finally, the corrected text can be re-presented to the reviewer for reevaluation, which will take another 2 weeks.

As a result, after about 3 months, we are able to send the text for publication in the upcoming issue of JTIT.

When the reviews are inconsistent, additional corrections are necessary, or the reviewer expects additional verification because the corrections ordered by the author are insufficient or additional problems arise, the review of the article may be extended by another month or more.

Editorial Work

Positively reviewed and corrected article is next prepared by the editorial office for publication. At the end of this process the author receives an copyedited version for approval.

Licensing

Manuscript submitted to JTIT should not be published or simultaneously submitted for publication elsewhere. By submitting a manuscript the author grants license to the National Institute of Telecommunications, for the use of the paper in the fields of exploitation: reproducing and fixing the paper, distributing the paper by means of introduction to trade, letting for use or rental of the original or copies, and distributing the paper by means of public exhibition, screening, presentation and broadcast as well as rebroadcast, and making the paper publicly available in such a manner that anyone could access it at a place and time selected thereby, or by making it available in a way not allowing selection of time or place, including by means of Internet or other networks.

Ghostwriting Declaration

We require formal declaration that the process of writing the paper was not influenced by any third party. In the article, all the contributions of other people are clearly indicated. The theories presented, methods used, analysis and research, as well as the copyrights to the drawings, photographs and other figures belong to the authors or are clearly credited in the text. The author must also indicate whether his work has received financial support and if the realization of the whole project was possible thanks to the permission and cooperation with scientific institutions, associations and others.

Other Information

- The JTIT being an Open Access Journal (OAJ) has no article processing charges (APCs). The published articles can be downloaded freely without payment.
- JTIT supports open access and using continuous publishing "publish-as-you-go" scheme. This means that we no longer wait to accumulate several articles into a quarterly issue before publication. Rather, articles are continuously added to current issues after acceptation. Publish-as-you-go reduces publication lag for our authors, and make the newest research available quickly. After completing the review process, an article is published online in the current issue with DOI registration. When the issue period ends, a new issue is activated. So accepted articles are published without waiting for the quarterly issue end.

Compressive Sensing-based Differential Channel Feedback Scheme Using Subspace Matching Pursuit Algorithm for B5G Wireless Systems

Baranidharan V and Surendar M 74

Optimizing Spectral and Energy Efficiency of Massive MIMO Networks Using MVO and API Algorithms

Bitat Hiba Ines, Maamri Fouzia, Khelfaoui Fatima, Djellab Hanane, et al. 81

National Institute of Telecommunications

Editorial Office

National Institute
of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland
https://www.gov.pl/web/instytut-lacznosci

phone +48 22 512 81 83
fax +48 22 512 84 00

e-mail: journal@jtit.pl
www.jtit.pl

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY 1/2025 •vol. 99